

MI3344

Aplikasi Desktop Berbasis Framework

Modul Praktikum

Hanya dipergunakan di lingkungan Fakultas Ilmu Terapan



**D3 Manajemen Informatika
Fakultas Ilmu Terapan
Universitas Telkom
2013**

Daftar Penyusun

1. Versi 0.1: 2013 01: Reza Budiawan, M.T.

Daftar Isi

Daftar Penyusun	i
Daftar Isi	ii
Daftar Nilai.....	iv
1 Bab I Pengenalan Netbeans Platform	1
1.1 Identitas	1
1.2 Pertanyaan Pendahuluan	2
1.3 Praktik	3
1.3.1 Exercise 1	3
1.4 Tes Akhir	10
1.5 Resume	10
1.5.1 Jurnal Pengamatan.....	10
1.5.2 Lain-lain.....	10
2 Bab II Menggunakan Dialog Pada Netbeans Platform	11
2.1 Identitas	11
2.2 Pertanyaan Pendahuluan	12
2.3 Praktik	13
2.3.1 Exercise 1: Pembuatan Wizard Dialog.....	13
2.4 Resume	18
2.4.1 Jurnal Pengamatan.....	18
3 Bab III Menggunakan Option Panel	19
3.1 Identitas	19
3.2 Pertanyaan Pendahuluan	20
3.3 Praktik	21
3.3.1 Exercise 1	21
3.4 Resume	25
3.4.1 Jurnal Pengamatan.....	25
4 Bab IV Menggunakan Help Module	26
4.1 Identitas	26
4.2 Pertanyaan Pendahuluan	27
4.3 Praktik	28

4.3.1	Exercise 1	28
4.4	Resume	31
4.4.1	Jurnal Pengamatan.....	31
5	Bab V Menggunakan BeanTreeView.....	32
5.1	Identitas	32
5.2	Pertanyaan Pendahuluan	33
5.3	Praktik	34
5.3.1	Exercise 1	34
5.4	Resume	38
5.4.1	Jurnal Pengamatan.....	38
6	Bab VI Koneksi Java Persistence API (JPA)	39
6.1	Identitas	39
6.2	Pertanyaan Pendahuluan	40
6.3	Praktik	41
6.3.1	Exercise 1	41
6.4	Resume	46
6.4.1	Jurnal Pengamatan.....	46
7	Bab VII Komponen Pelengkap.....	47
7.1	Identitas	47
7.2	Praktik	48
7.2.1	Exercise 1	48
7.2.2	Jurnal Pengamatan.....	51
	Daftar Pustaka	52

Daftar Nilai

Nim – Nama :

Kelas :

Semester/Tahun Ajaran:

Nomor Modul	TP	Jurnal	TA	Nilai per Modul	Paraf-Nama Asisten
1					
2					
3					
4					
5					
6					
7					
Nilai Akhir Praktikum					

Ket:

Prosentase nilai TP, Jurnal dan TA disesuaikan dengan aturan praktikum yang berlaku.

TP = Tugas Pendahuluan

TA = Tes Akhir

1 BAB I PENGENALAN NETBEANS PLATFORM

1.1 IDENTITAS

Kajian

Pengenalan Framework, Netbeans Platform Application, Rich Client.

Topik

1. Windowing Netbeans Platform
2. Aksi Windowing Netbeans Platform

Kompetensi Utama

1. Membuat aplikasi GUI sederhana menggunakan komponen window Netbeans Platform
2. Mampu menambahkan aksi kepada window Netbeans Platform

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 2 x 100 menit

Parameter Penilaian

1. Jurnal
2. Tugas Akhir

1.2 PERTANYAAN PENDAHULUAN

Kerjakan tugas pendahuluan ini, jika sudah selesai, kumpulkan kepada asisten anda. Waktu pengerjaan maksimal 15 menit.

1. Jelaskan sekilas tentang Netbeans Platform?
2. Komponen apa saja yang terdapat di Netbeans Platform?
3. Bagaimana struktur folder dan konfigurasi dari Netbeans Platform?

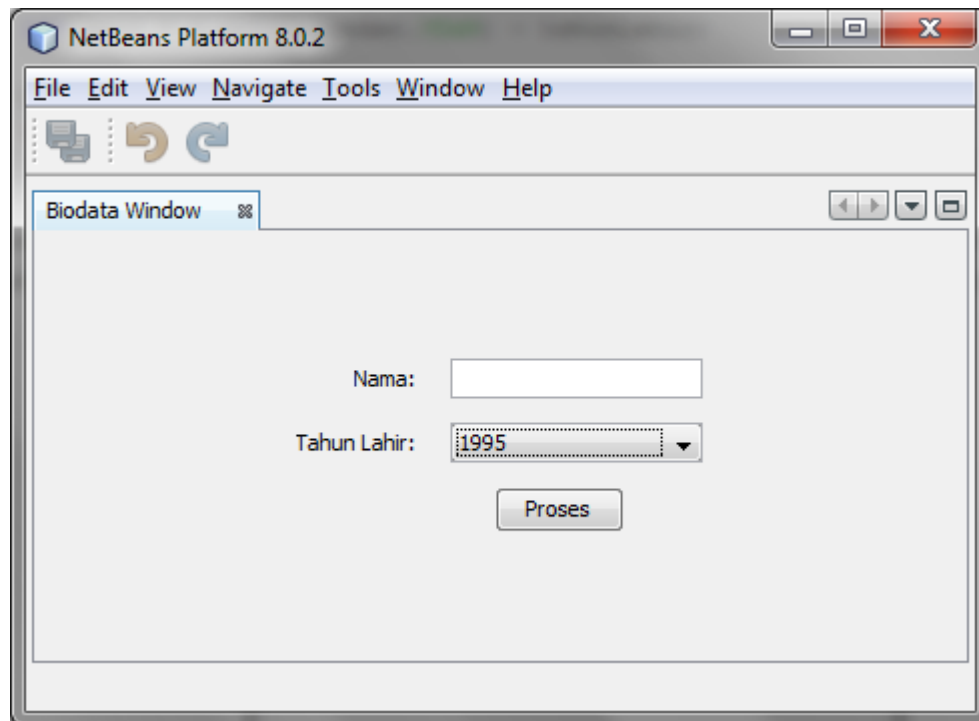
1.3 PRAKTIK

1.3.1 Exercise 1

Pada bagian ini, akan dipelajari cara untuk membuat aplikasi dengan menggunakan Netbeans Platform.

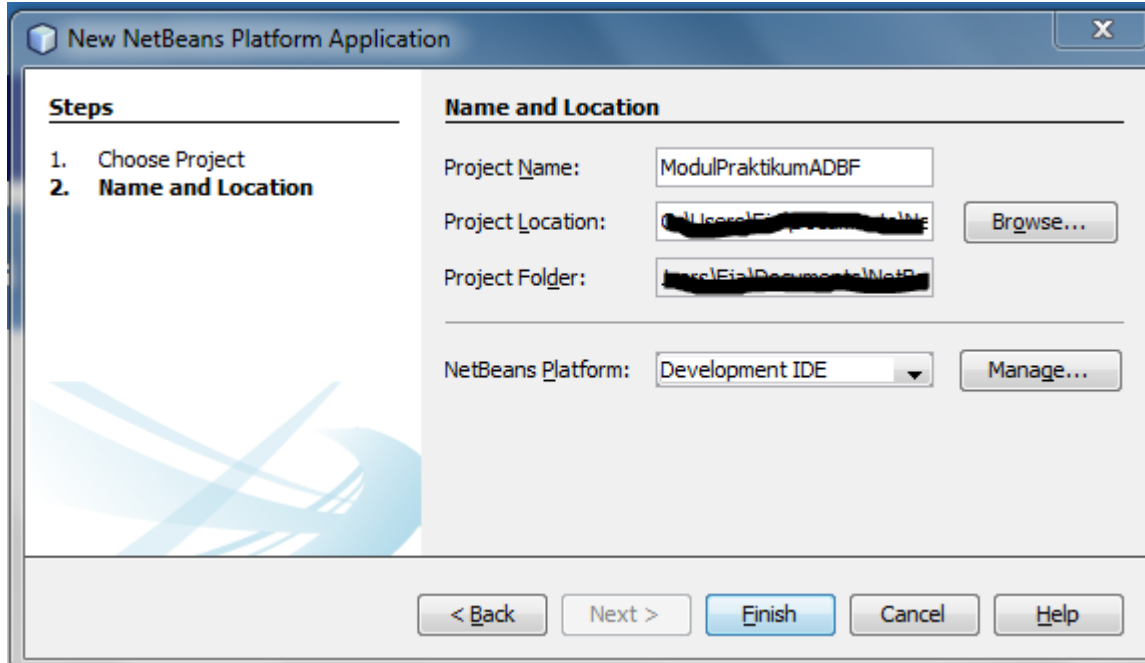
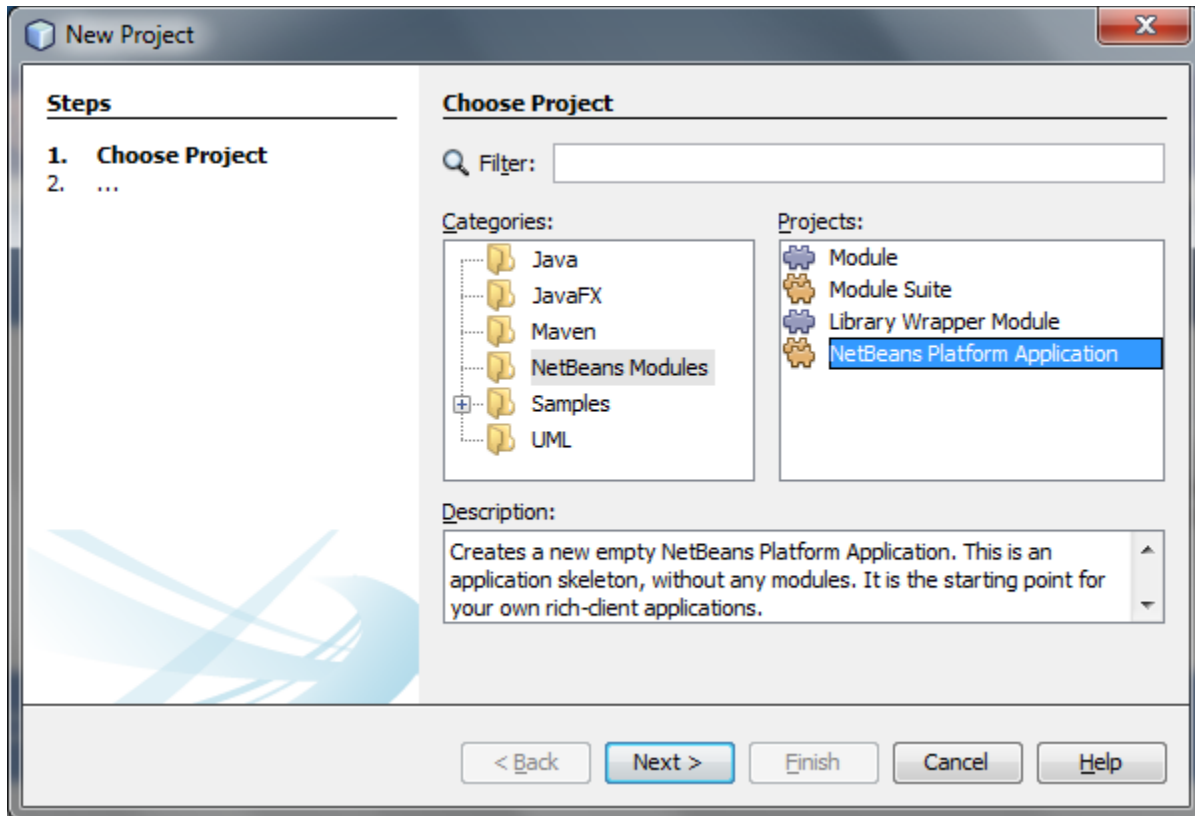
1.3.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap **Permasalahan/Soal**

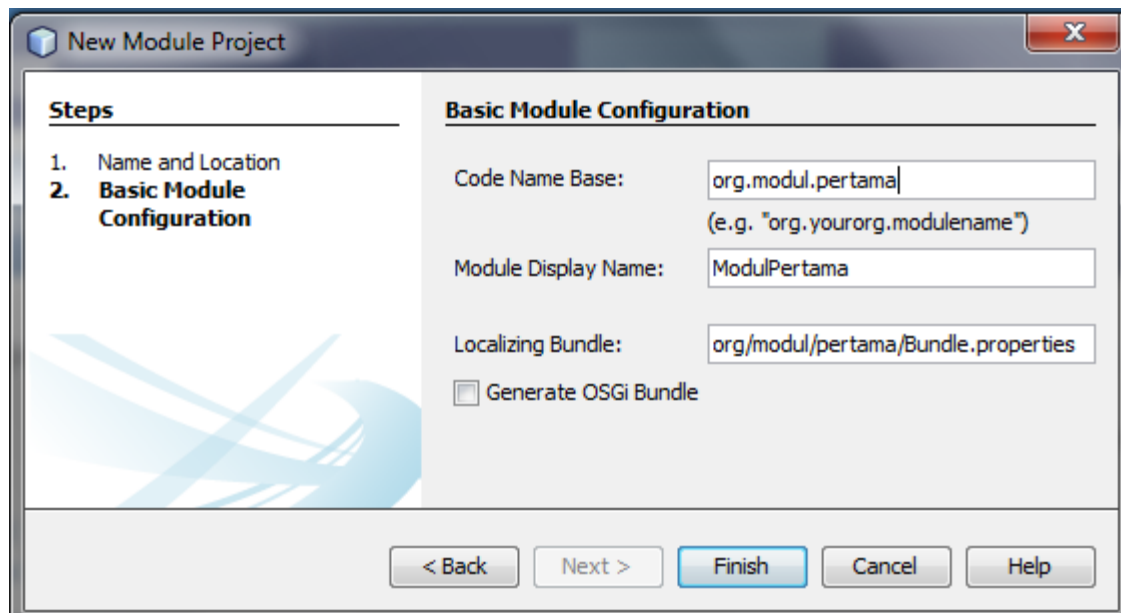
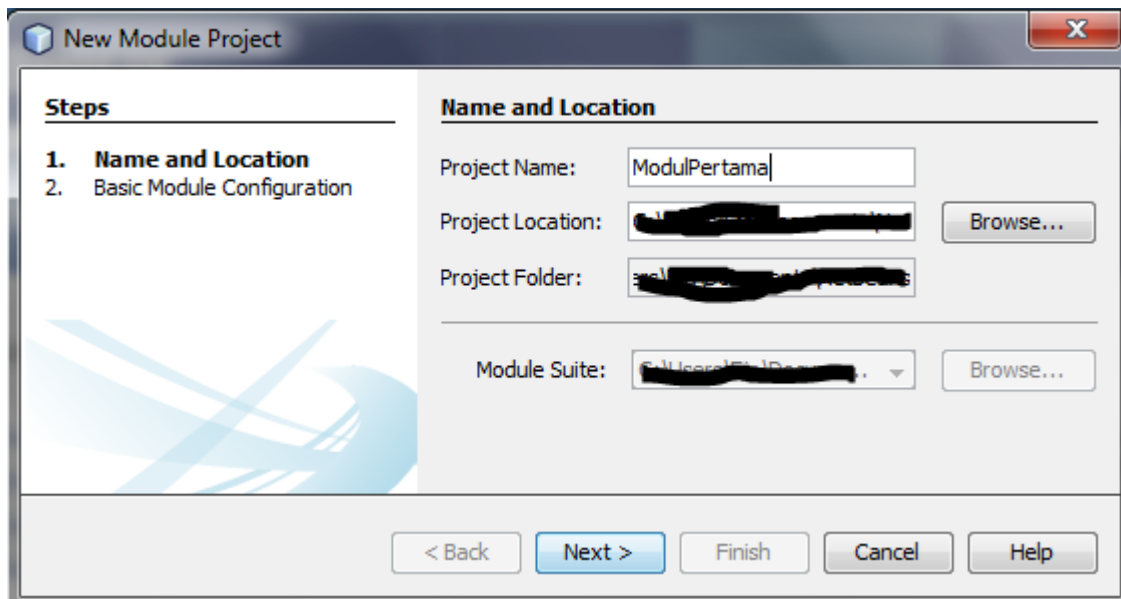
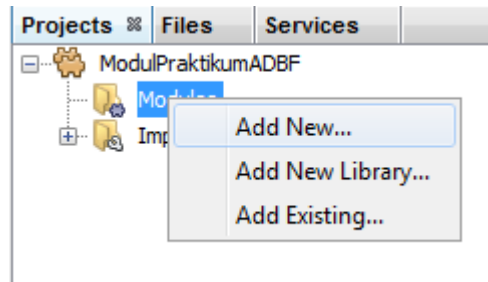
Buatlah sebuah aplikasi sederhana yang meminta 2 masukan dan mengolahnnya dengan menggunakan Netbeans Platform. Hasil akhir diperlihatkan sebagai berikut:



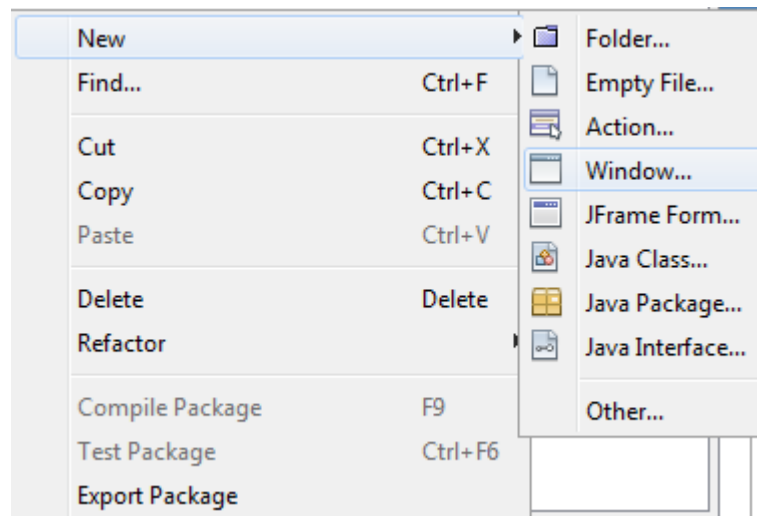
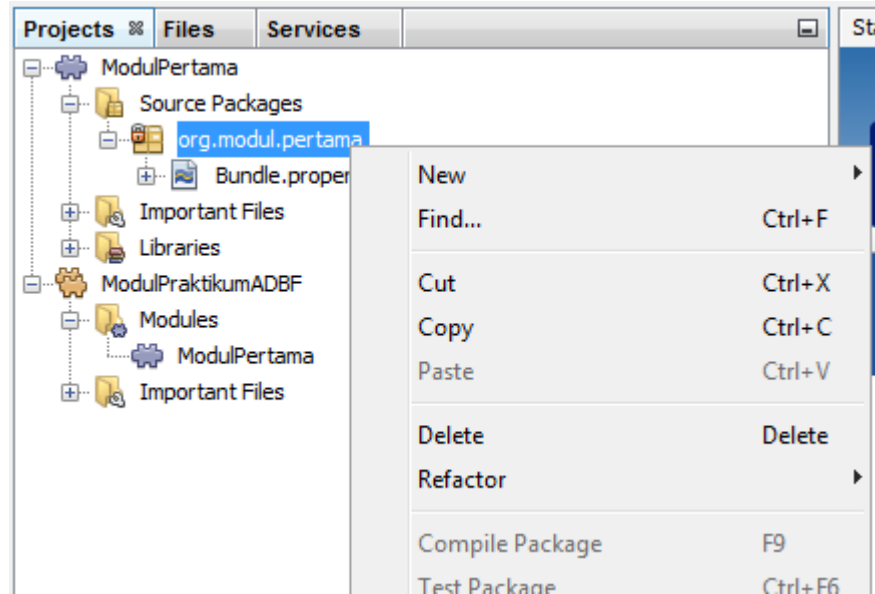
Langkah Penyelesaian:

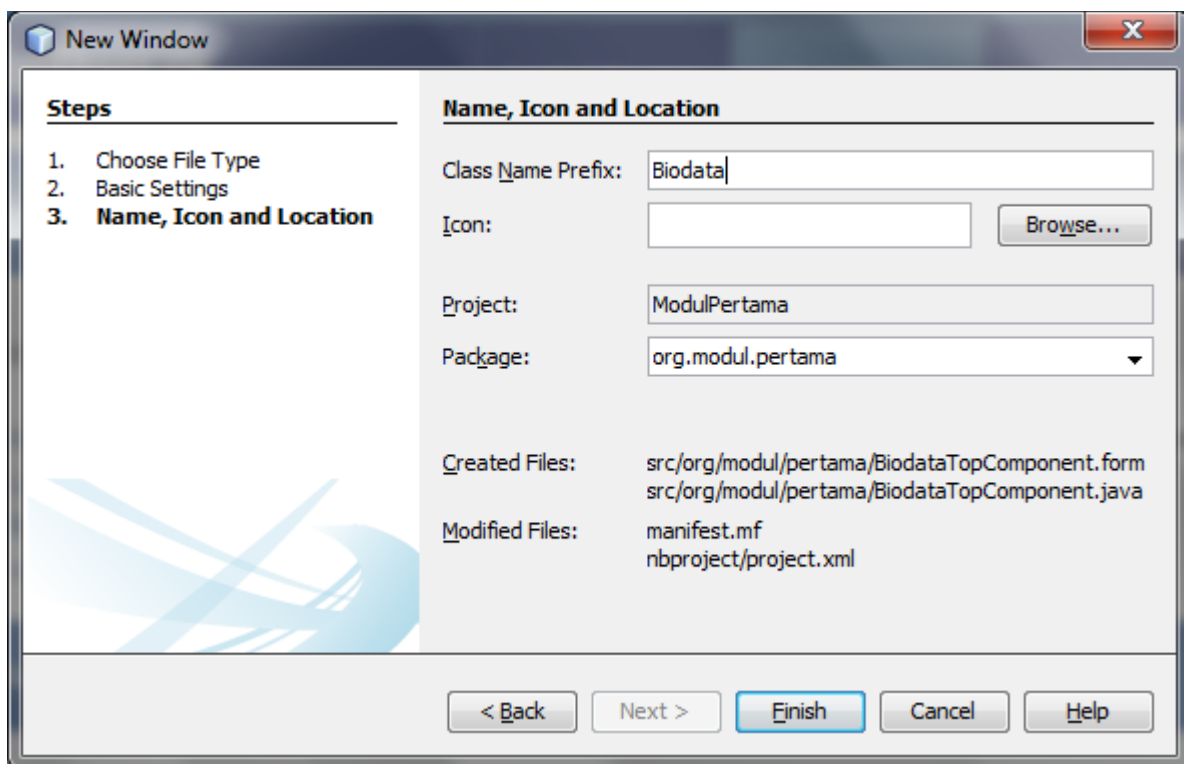
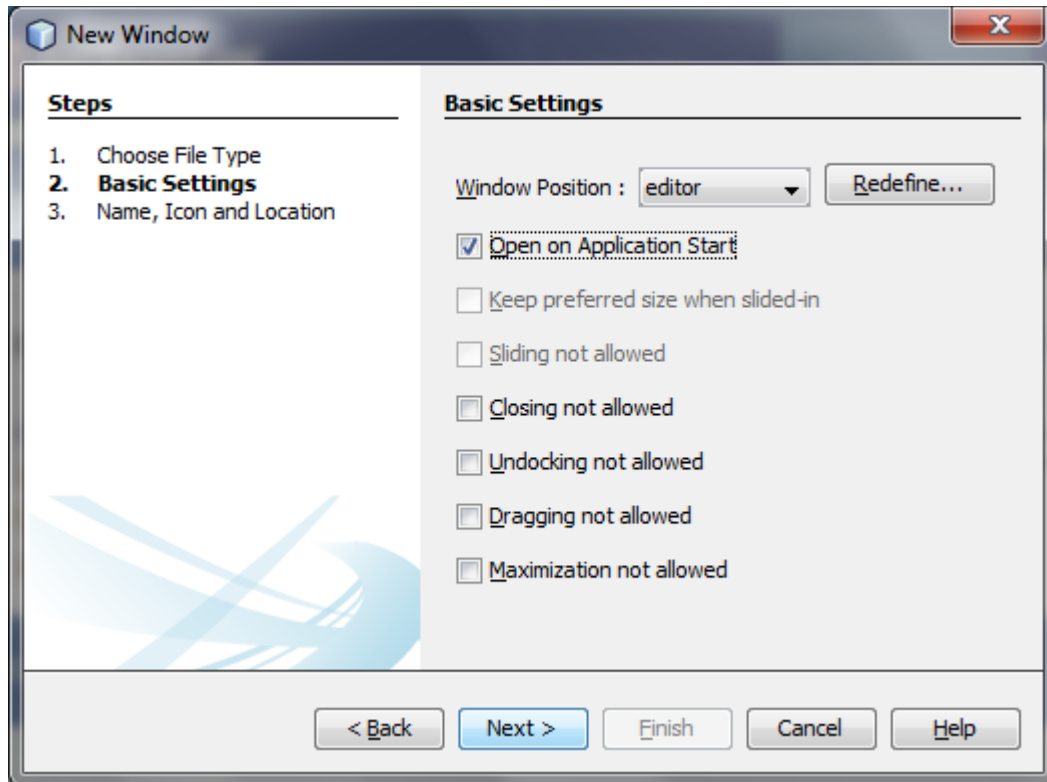
Langkah pertama, buat sebuah Netbeans Platform Application (NPA) melalui IDE Netbeans. Setelah membuat NPA (nama bebas), tambahkan 1 "Module" (nama bebas) pada NPA tersebut.





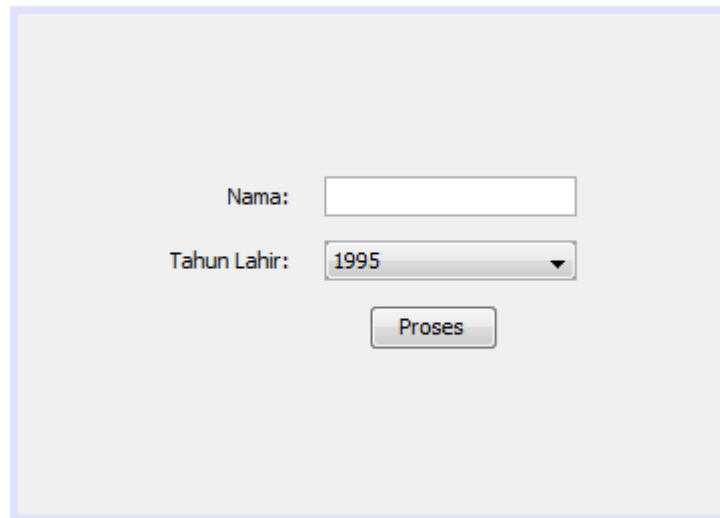
Pada modul yang sudah dibuat sebelumnya (ModulPertama), buat sebuah TopComponent Window pada package yang telah tersedia. Berikan konfigurasi agar Window terletak pada posisi editor, dan terbuka pada saat pertama kali program dijalankan (Open on Application Start).





Berikan Icon jika diperlukan (perhatikan letak Icon nantinya), dan perhatikan file yang terbentuk oleh framework yang digunakan, serta file yang dimodifikasi isinya. Setelah menekan Finish, atur agar kanvas

desain menampilkan komponen seperti yang ditampilkan pada gambar di bawah ini. Pastikan tidak menggunakan `NullLayout` atau `AbsoluteLayout`.

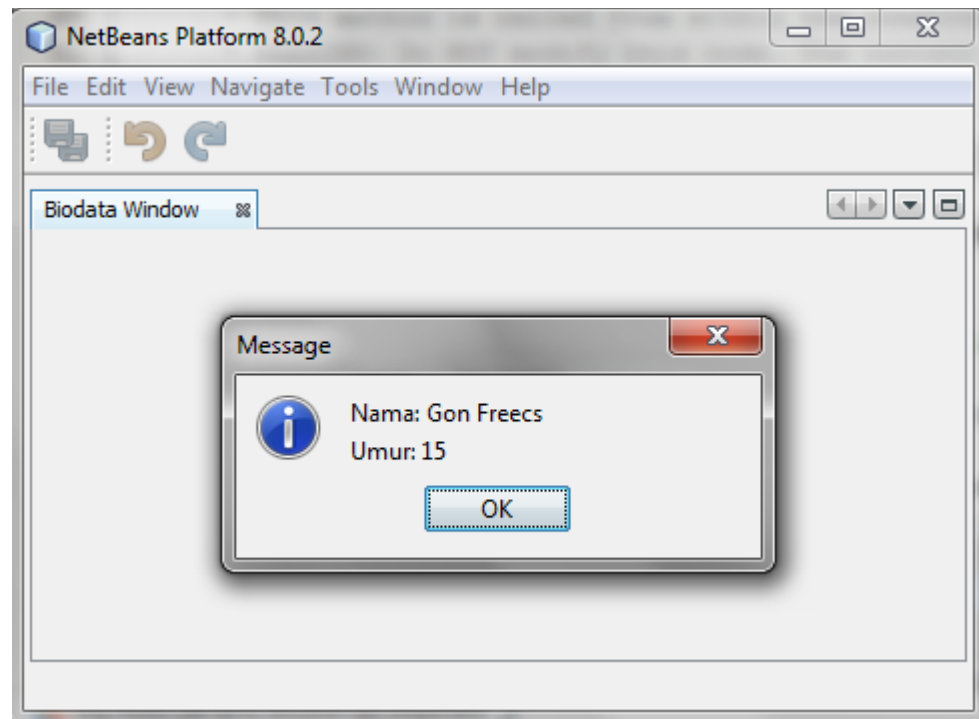
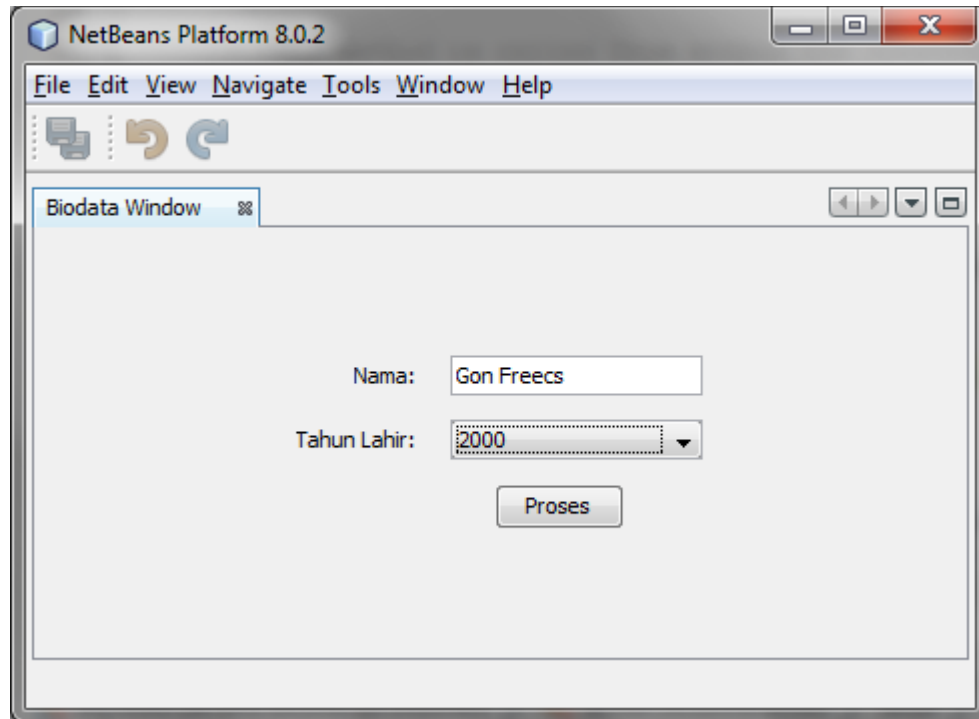
A screenshot of a Java Swing window with a light gray background. It contains a form with three components: a text field labeled 'Nama:' with an empty input box; a dropdown menu labeled 'Tahun Lahir:' with '1995' selected and a downward arrow; and a button labeled 'Proses' centered below the dropdown.

Berikan aksi pada tombol sebagai berikut:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //ambil nama:  
    String nama = jTextField1.getText();  
    int tahunLahir = Integer.parseInt(jComboBox1.getSelectedItem().toString());  
    Calendar c = Calendar.getInstance();  
    int umur = c.get(Calendar.YEAR) - tahunLahir;  
  
    String tampil = "Nama: "+nama+"\nUmur: "+umur;  
    JOptionPane.showMessageDialog(this, tampil);  
}
```

Jangan lupa untuk meng-*import* package terkait.

Jalankan NPA dengan klik kanan pada project NPA, pilih "Run", atau dengan menekan segitiga kanan berwarna hijau pada toolbar IDE Netbeans (pastikan project NPA merupakan project utama). Hasil akhir diperlihatkan sebagai berikut:



1.3.1.2 *Praktikum Mandiri*

Tambahkan sebuah informasi berupa jenis kelamin (radio button), dan tampilkan kembali sebagai hasil eksekusi "Proses". Jika tidak terjadi perubahan, lakukan "Clean and Build" dari NPA. Perhatikan, jika ada perubahan nama Window, di bagian kode mana yang perlu dilakukan perubahan?

1.4 TES AKHIR

Tambahkan sebuah Window kembali yang meminta perhitungan 2 buah bilangan, perhitungan terdiri dari operasi tambah, kurang, kali, bagi, div, dan mod. Pengaturan letak komponen bebas, usahakan tidak menggunakan NullLayout atau AbsoluteLayout.

1.5 RESUME

1.5.1 Jurnal Pengamatan

Selesaikan contoh kasus, kemudian jawablah pertanyaan-pertanyaan yang terdapat pada bagian pengamatan yang ada di bawah kode soal lengkapnya.

1.5.2 Lain-lain

1. Kumpulkan jawaban anda kepada para asisten sebelum praktikum diakhiri.
2. Media pengerjaan dan pengumpulan bebas, ikuti aturan yang telah ditetapkan oleh asisten praktikum.

2 BAB II MENGGUNAKAN DIALOG PADA NETBEANS PLATFORM

2.1 IDENTITAS

Kajian

Komponen Netbeans Platform

Topik

Wizard Dialog

Kompetensi Utama

1. Mampu membangun sebuah aplikasi desktop menggunakan Wizard
2. Memahami properties dari komponen Wizard

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 2 x 100 menit

Parameter Penilaian

1. Jurnal Pengamatan

2.2 PERTANYAAN PENDAHULUAN

1. Apakah yang dimaksud dengan Wizard Dialog?
2. Sebutkan jenis-jenis Wizard Dialog pada Netbeans Platform, dan cara membuatnya.

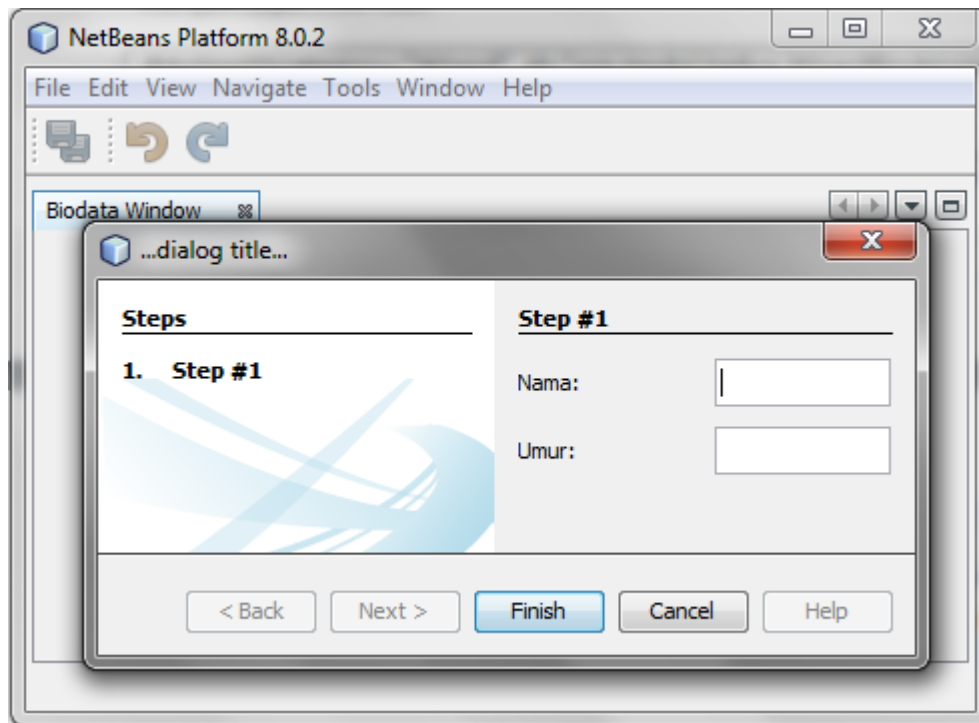
2.3 PRAKTIK

2.3.1 Exercise 1: Pembuatan Wizard Dialog

Pada bagian ini, akan dipelajari mengenai penggunaan Wizard Dialog dan karakteristiknya.

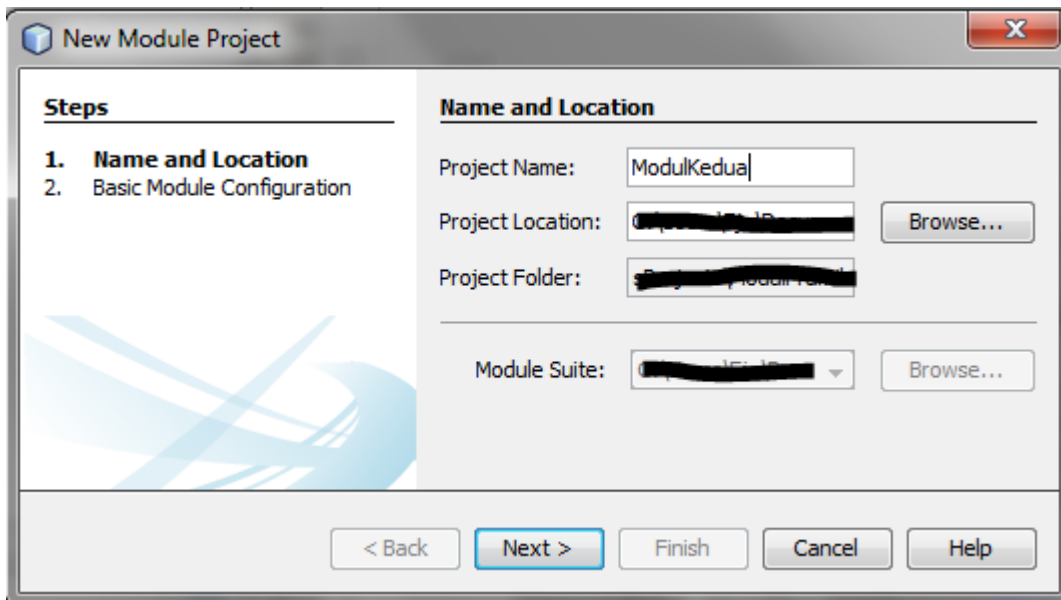
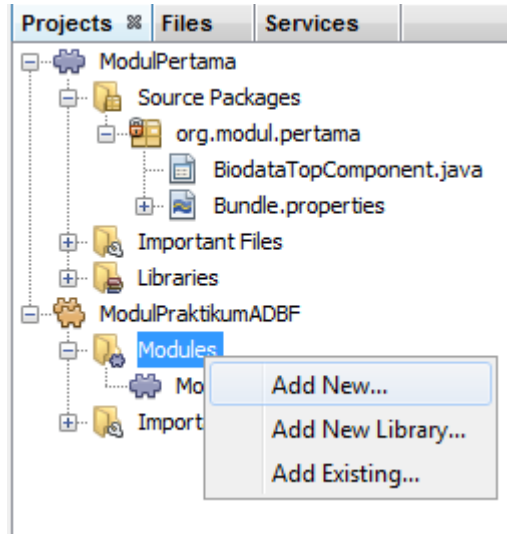
2.3.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap Permasalahan/Soal

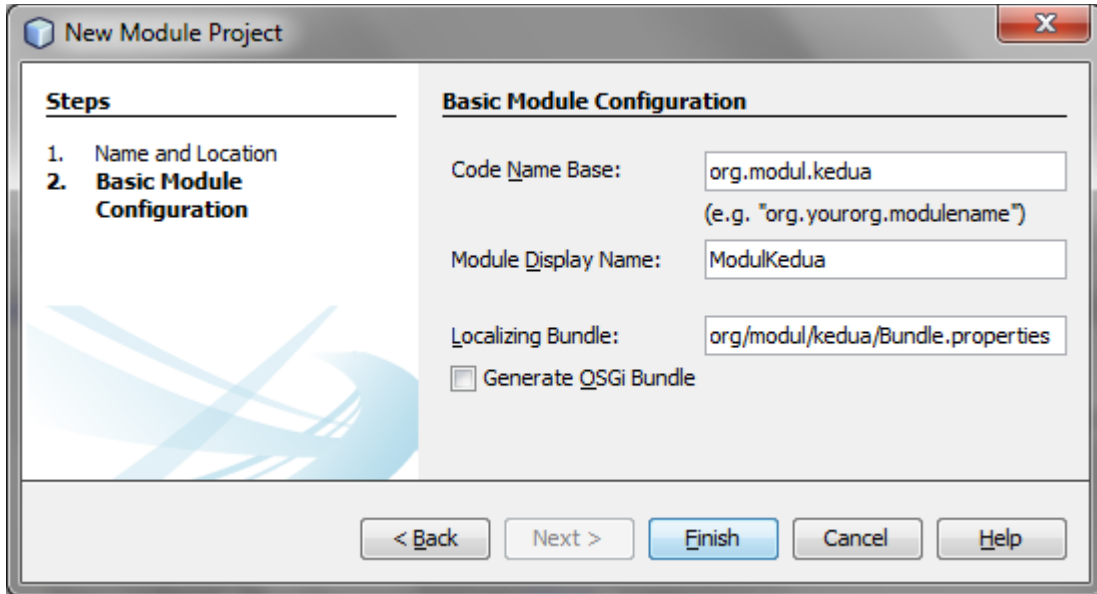
Buatlah aplikasi NPA berdasarkan pada modul sebelumnya. Tambahkan modul untuk menambahkan komponen Wizard Dialog pada NPA tersebut.



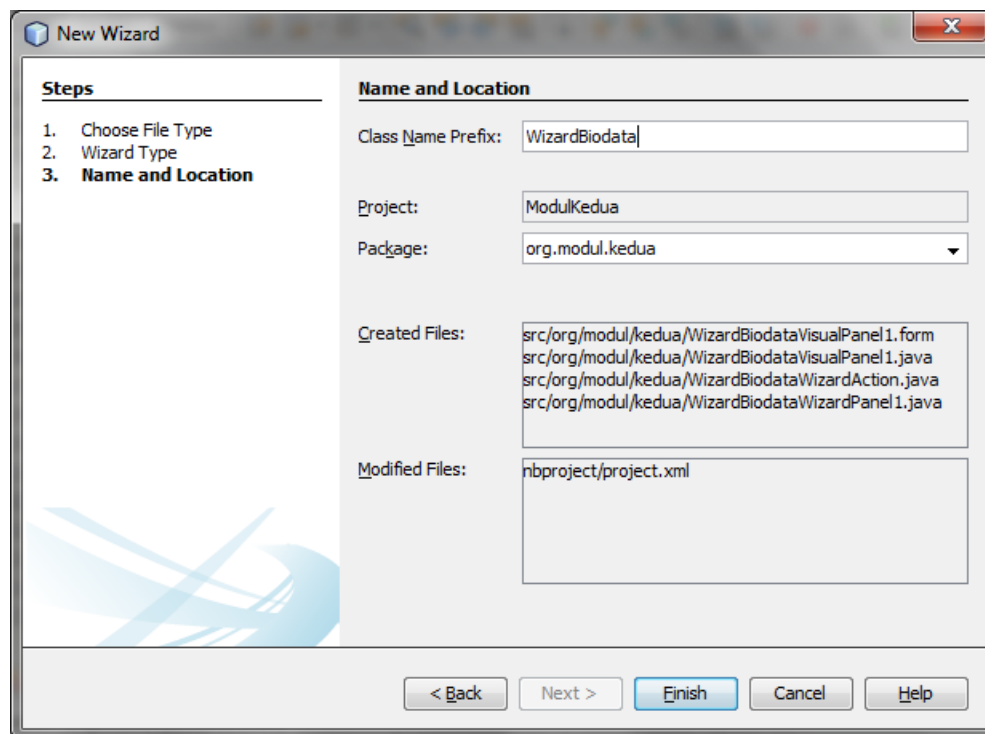
Langkah Penyelesaian

Pada NPA sebelumnya, buat sebuah module. Pada module ini akan dimasukkan komponen Wizard Dialog. Pemisahan module dari module sebelumnya bersifat opsional. Jika ingin menggunakan module sebelumnya, hal ini masih diperbolehkan.





Tambahkan sebuah “Wizard” pada package modul yang baru di-generate. Berikan konfigurasi “Custom” pada “Configuration Type”, dan “Static” pada Wizard Step Sequence. Setelah itu, berikan “1” untuk “Number Wizard of Panels”. Tekan Next, dan beri nama “WizardBiodata” (nama bersifat bebas). Perhatikan bahwa terdapat 4 file yang terbentuk dari langkah-langkah yang sudah dilakukan.



Penjelasan dari setiap file adalah sebagai berikut:

1. WizardBiodataVisualPanel1.form & WizardBiodataVisualPanel1.java: sebagai tampilan Wizard

2. WizardBiodataWizardAction.java: sebagai tempat pengaturan konfigurasi Wizard yang dibentuk secara keseluruhan
3. WizardBiodataWizardPanel1.java: sebagai tempat konfigurasi bagi WizardBiodataVisualPanel1

Setelah 3 file tersebut terbentuk, desain kanvas dari “WizardBiodataVisualPanel1.java” seperti gambar di bawah ini. Hindari men-desain kanvas dengan GroupLayout atau AbsoluteLayout. Contoh di bawah menggunakan GridLayout.

Melalui Wizard, data masukan pengguna dapat disimpan dalam objek tertentu (melalui konfigurasi Wizard NPA). Hal itulah yang akan dilakukan di sini. Nama dan Umur akan tersimpan dalam sebuah objek agar dapat digunakan kembali oleh pengguna untuk keperluan apa pun.

Untuk melakukan penyimpanan, yang diperlukan adalah beberapa fungsi get untuk mengambil nilai dari 2 JTextField. Kedua get ini dituliskan pada class yang sama pada saat mendeklarasikan dan membentuk JTextField.

```
public String getNama(){
    return jTextField1.getText();
}

public String getUmur(){
    return jTextField2.getText();
}
```

Pindah ke file “WizardBiodataWizardPanel1.java”, ambil nilai dari getNama() dan getUmur(). Simpan ke dalam sebuah variabel melalui method storeSetting().

```
wiz.putProperty("nama", component.getNama());
wiz.putProperty("umur", component.getUmur());
```

Variabel objek component mengacu pada atribut yang dimiliki oleh class “WizardBiodataWizardPanel1”. Pengisian component ini berada pada method “getComponent()”. Nilai yang sudah disimpan dapat dipanggil menggunakan getProperty(). Pemanggilan ini dapat dilakukan di akhir setelah menyelesaikan semua aksi dari Wizard.

Agar dapat ditampilkan kembali, berikan baris kode berikut pada class “WizardBiodataWizardAction.java”.

```
String nama = wiz.getProperty("nama").toString();
String umur = wiz.getProperty("umur").toString();
String tampil = "Nama: "+nama+"\nUmur: "+umur;
JOptionPane.showMessageDialog(null, tampil);
```

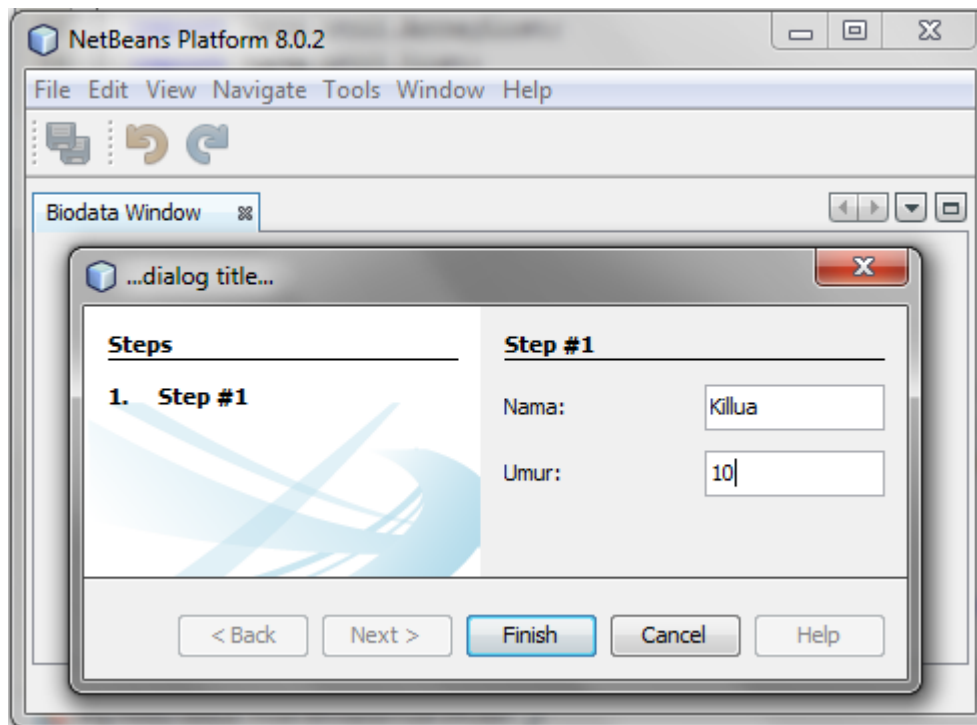
Peletakan kode pada bagian pengecekan FINISH_OPTION pada Wizard yang dibentuk.

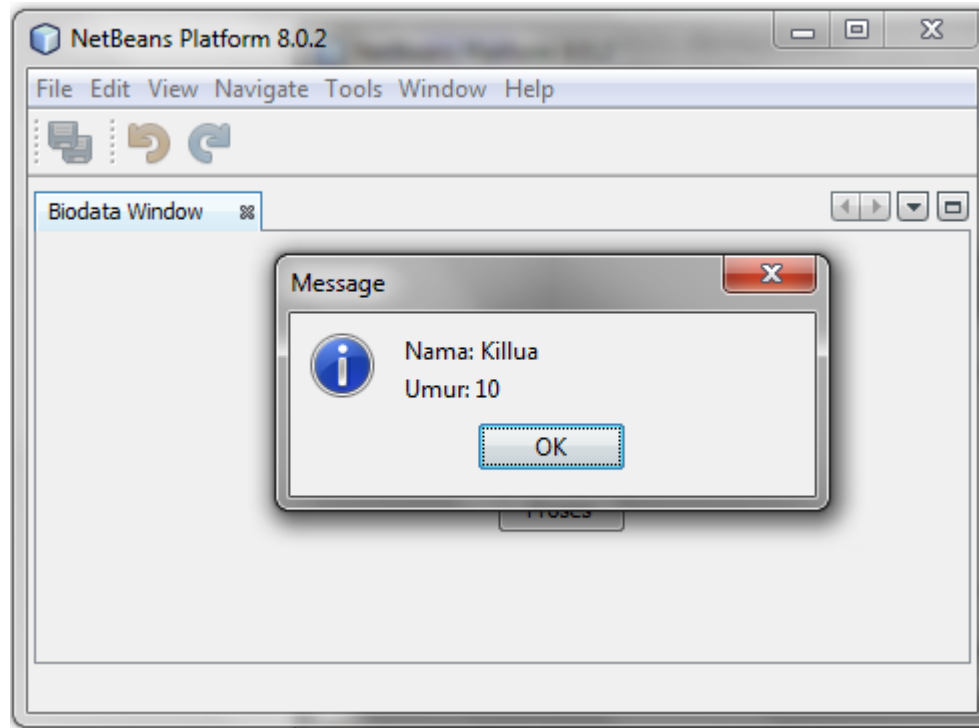
```
if (DialogDisplayer.getDefault().notify(wiz) == WizardDescriptor.FINISH_OPTION) {
    String nama = wiz.getProperty("nama").toString();
    String umur = wiz.getProperty("umur").toString();
    String tampil = "Nama: "+nama+"\nUmur: "+umur;
    JOptionPane.showMessageDialog(null, tampil);
}
```

Wizard dapat dieksekusi. Langkah berikutnya, membuat aksi untuk memanggil Wizard yang sudah dapat dieksekusi. Pada "WizardBiodataWizardAction", terdapat 3 anotasi yang diberi penanda "//". Hilangkan penanda tersebut untuk anotasi "ActionID", "ActionRegistration", dan "ActionReference". Ubah nilainya menjadi seperti berikut:

```
@ActionID(category="Wizard", id="org.modul.kedua.WizardBiodataWizardAction")
@ActionRegistration(displayName="Wizard Biodata")
@ActionReference(path="Menu/Tools", position=1)
@ActionID(category="Wizard", id="org.modul.kedua.WizardBiodataWizardAction")
@ActionRegistration(displayName="Wizard Biodata")
@ActionReference(path="Menu/Tools", position=1)
```

Jalankan NPA, dan pada menu "Tools" terdapat "Wizard Biodata". Aksi ini akan membuka Wizard yang sudah dibuat sebelumnya.





2.3.1.2 *Praktik Mandiri*

Ubahlah nama dari *title* Wizard, lalu tambahkan background pada bagian kiri Wizard. Hal apa yang harus dimodifikasi? Bagaimana agar Action dari Wizard dapat dilihat pada menu File, bukan Tools?

2.3.1.3 *Praktik Mandiri*

Wizard terdiri dari *dynamic* Wizard dan *static* Wizard. Buatlah sebuah dynamic Wizard, lalu jalankan dengan scenario tertentu. Kasus dan scenario dari Wizard dibebaskan.

2.4 RESUME

2.4.1 **Jurnal Pengamatan**

Selesaikan contoh kasus, kemudian jawablah pertanyaan-pertanyaan yang terdapat pada bagian pengamatan yang ada di bawah kode soal lengkapnya.

3 BAB III MENGGUNAKAN OPTION PANEL

3.1 IDENTITAS

Kajian

Komponen Netbeans Platform

Topik

Option Panel

Kompetensi Utama

1. Mampu membangun sebuah aplikasi desktop menggunakan Option Panel
2. Memahami properties dari komponen Option Panel

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 2 x 100 menit

Parameter Penilaian

1. Jurnal Pengamatan

3.2 PERTANYAAN PENDAHULUAN

1. Sebutkan jenis-jenis dari Panel, dan apakah perbedaannya?
2. Bagaimana cara menyimpan isi masukan pengguna menggunakan NbPreferences?
3. Apa fungsi dari NbPreferences itu sendiri?

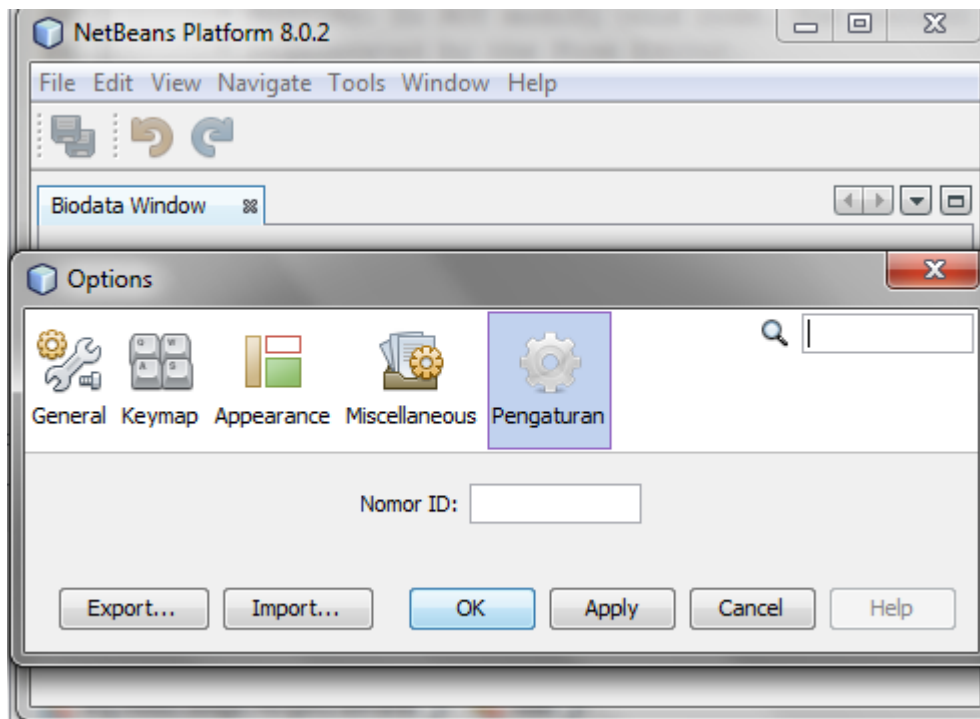
3.3 PRAKTIK

3.3.1 Exercise 1

Pada bagian ini, akan dipelajari tentang cara menggunakan Panel (Primary) sebagai komponen dari Netbeans Platform Applicaton.

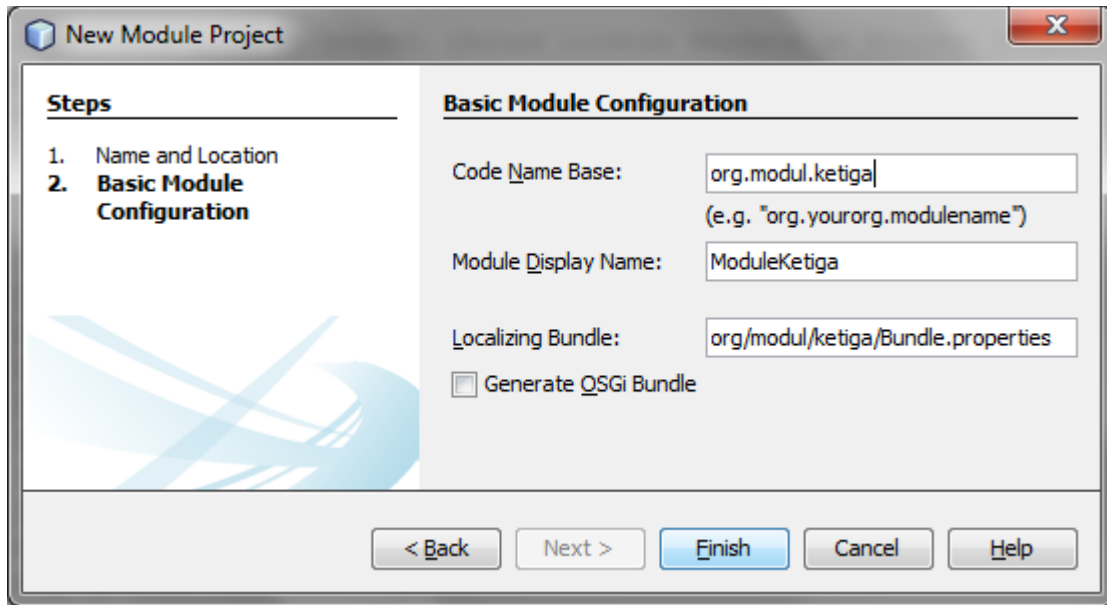
3.3.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap Permasalahan/Soal

Buatlah sebuah aplikasi NPA (berdasarkan modul sebelumnya), dan tambahkan sebuah primary Panel sebagai berikut:

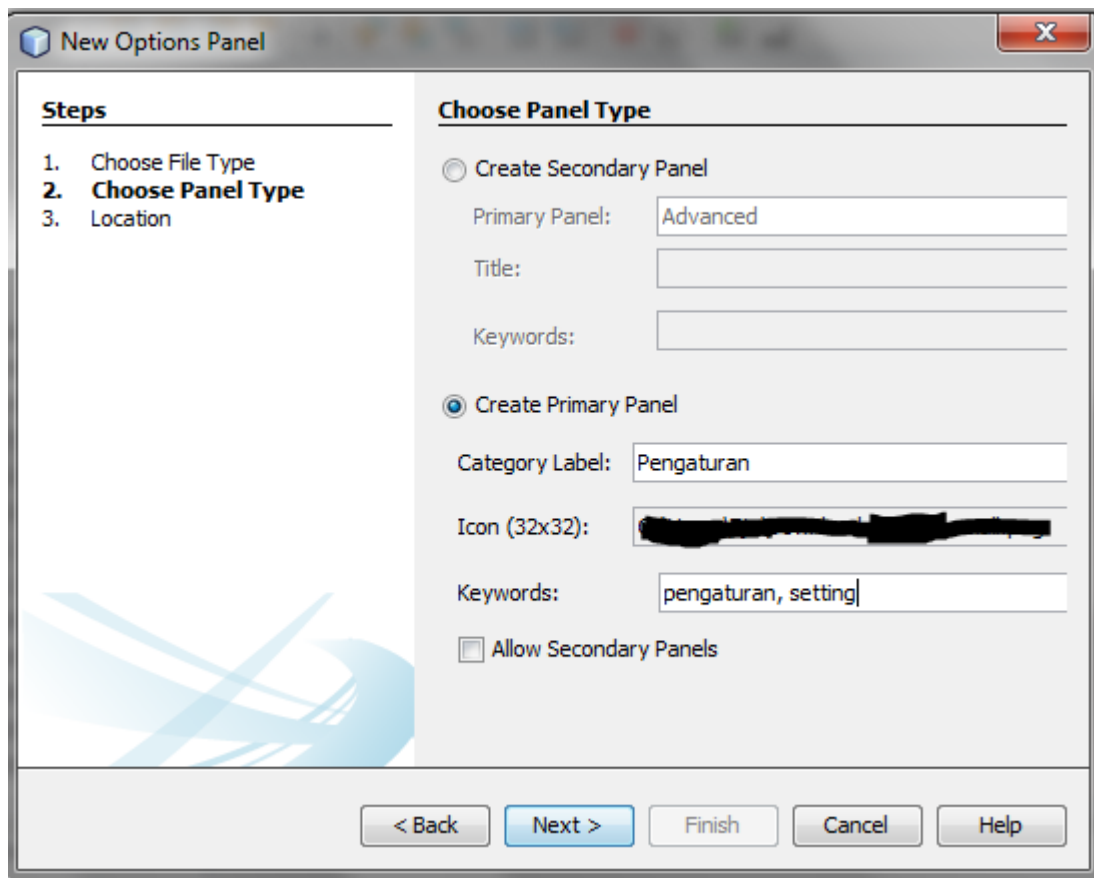


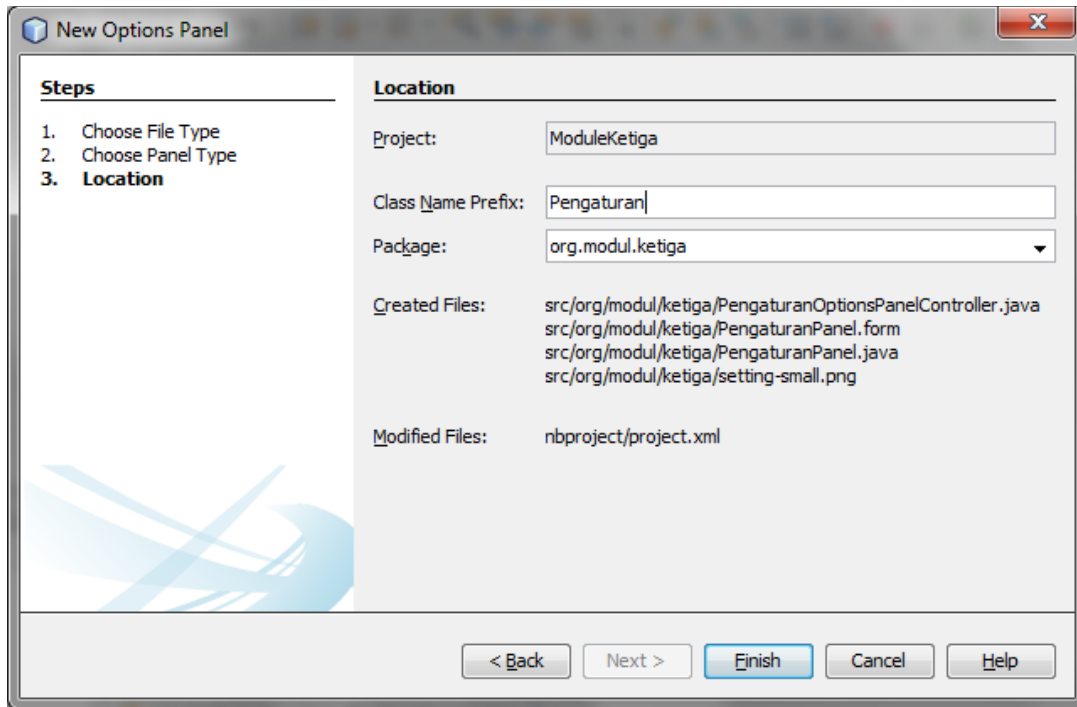
Langkah Penyelesaian

Buat sebuah module baru dari NPA sebelumnya (module baru bersifat optional).



Buat sebuah Option Panel pada module yang baru dibentuk. Perhatikan *files* yang terbentuk.





Desain kanvas yang ada dengan pengaturan sebagai berikut. Hindari penggunaan `NullLayout` dan `AbsoluteLayout`.

Masukan dari pengguna berupa Nomor ID, dapat disimpan ke dalam sebuah file sebagai acuan konfigurasi yang digunakan oleh NPA yang dihasilkan nantinya. Cara melakukan penyimpanan dapat dengan memodifikasi “`PengaturanPanel.java`”, pada method `store()`. Sedangkan untuk menempatkan nilai yang telah disimpan, dapat memodifikasi method “`load()`”. Penyimpanan ke dalam *file* dapat dilakukan dengan menggunakan syntax:

```
NbPreferences.forModule(<class acuan>).put(<nama var>, <teks yg disimpan>);
```

Pada package “`org.openide.util.NbPreferences`”. Sehingga, pada class ini (method “`store()`”) dapat ditambahkan kode berikut:

```
NbPreferences.forModule(PengaturanPanel.class).put("user id", jTextField1.getText());
```

```

void store() {
    NbPreferences.forModule(PengaturanPanel.class).put("user id", jTextField1.getText());
}

```

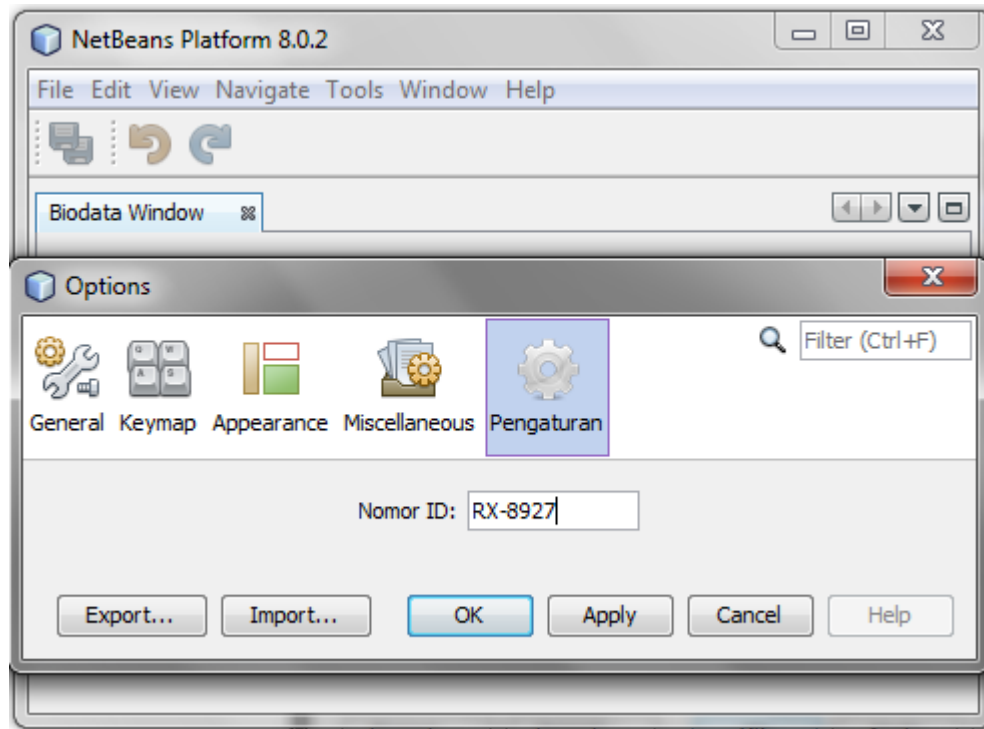
Sedangkan pada method load(), tambahkan kode berikut:

```

String nama = NbPreferences.forModule(PengaturanPanel.class).get("user id", "");
jTextField1.setText(nama);

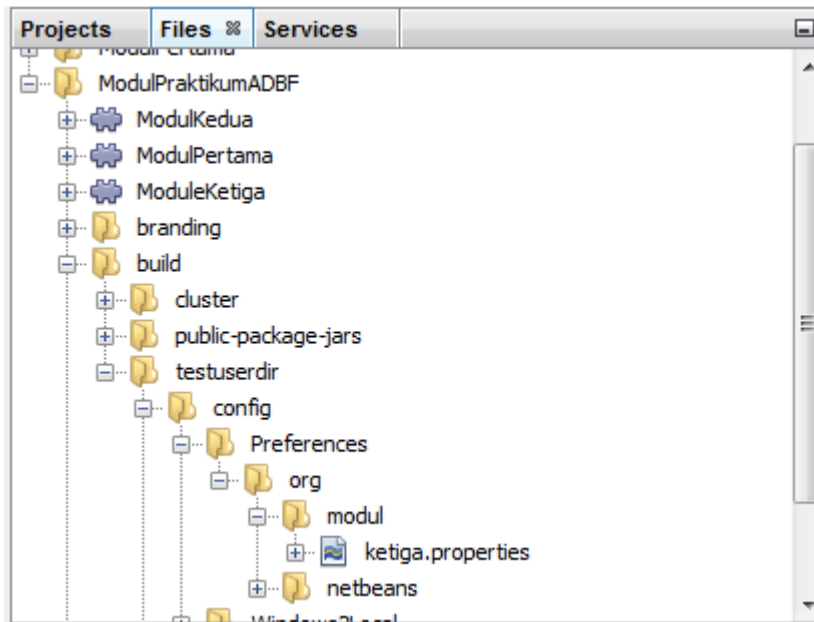
```

Eksekusi NPA, dan berikut hasil dari eksekusi Option Panel.



Nilai "RX-8927" akan dapat dilihat kembali oleh pengguna ketika memilih Tools → Option → Pengaturan kembali. Hal ini dikarenakan method "load" telah dimodifikasi. Nilai ini juga dapat dilihat pada bagian "build" dari NPA. Nilai terdapat pada file "Ketiga.properties". Isi dari file tersebut adalah sebagai berikut:

```
user\id=RX-8927
```



3.3.1.2 *Praktik Mandiri*

Buatlah sebuah Primary Panel dan Sebuah Secondary Panel (dengan 2 sub panel). Skenario dan layout dibebaskan.

3.4 RESUME

3.4.1 **Jurnal Pengamatan**

Selesaikan contoh kasus, kemudian jawablah pertanyaan-pertanyaan yang terdapat pada bagian pengamatan yang ada di bawah kode soal lengkapnya.

4 BAB IV MENGGUNAKAN HELP MODULE

4.1 IDENTITAS

Kajian

Komponen Netbeans Platform

Topik

Help Module

Kompetensi Utama

1. Mampu membangun sebuah aplikasi desktop menggunakan dokumentasi Help
2. Memahami properties dari komponen Help Module

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 2 x 100 menit

Parameter Penilaian

1. Jurnal Pengamatan

4.2 PERTANYAAN PENDAHULUAN

1. Apakah yang dimaksud dengan Help Module pada NPA?
2. Bagaimana aturan penulisan standar dari file XML?
3. File apa saja yang aan digunakan pada Help Module?

4.3 PRAKTIK

4.3.1 Exercise 1

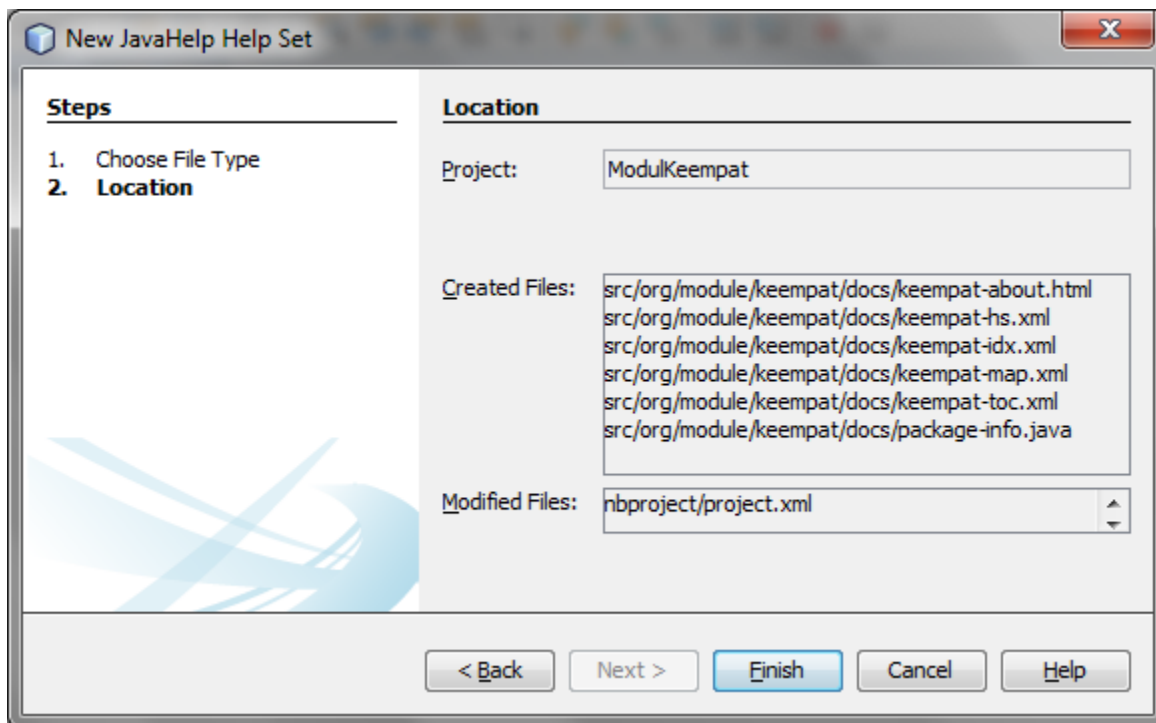
Pada bagian ini akan dibahas cara membuat dokumentasi Help pada Netbeans Platform Application.

4.3.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap **Permasalahan/Soal**

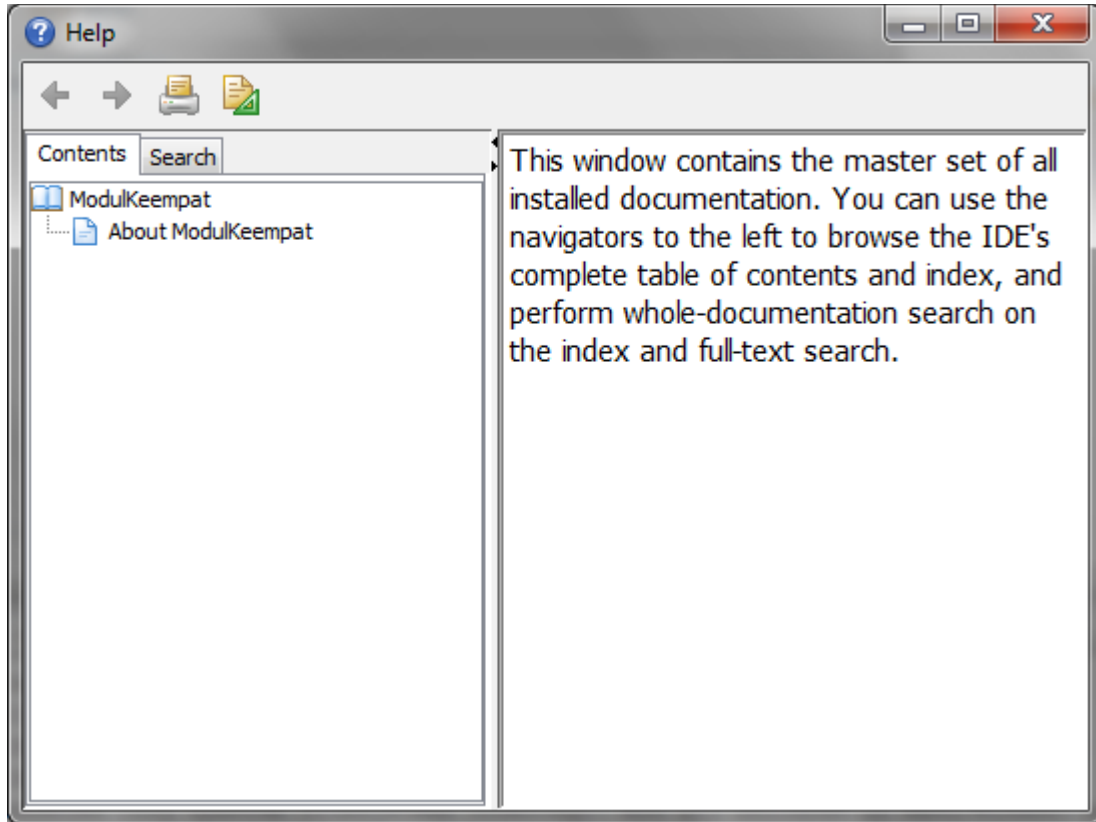
Buatlah dokumentasi Help dari aplikasi NPA yang telah dibuat sebelumnya.

Langkah Penyelesaian

Dari project NPA sebelumnya, buat satu modul kembali. Lalu tambahkan “JavaHelp Help Set”. Perhatikan *files* yang terbentuk.



Jika dijalankan NPA yang telah ditambahkan Help Set, akan tampil hal berikut pada bagian Help → “Help Contents”.



Untuk menambahkan satu halaman baru, buat sebuah file HTML seperti kode di bawah ini (halamanSatu.html).

```
<html>
  <head>
    <title>Dokumentasi Pertama</title>
    <link rel="stylesheet"
href="nbdocs://org.netbeans.modules.usersguide/org/netbeans/modules/usersguide/ide.css"
type="text/css">

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h2>1. Pengenalan</h2>
    <div>Dokumentasi ini dituliskan pertama kali</div>
  </body>
</html>
```

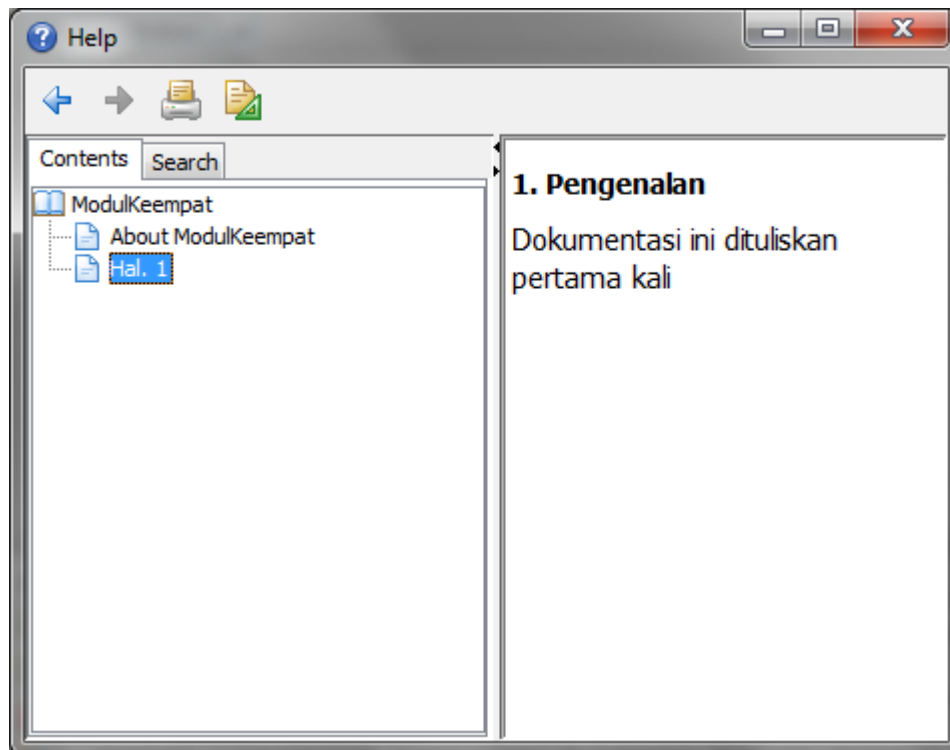
Berikutnya, berikan pemetaan antara url dengan target yang ditentukan. Hal ini dilakukan pada keempat-map.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Map Version 2.0//EN"
"http://java.sun.com/products/javahelp/map_2_0.dtd">
<map version="2.0">
  <mapID target="org.module.keempat.about" url="keempat-about.html"/>
  <mapID target="org.module.keempat.halSatu" url="halamanSatu.html"/>
</map>
```

Berikan pemetaan terhadap node (sebelah kiri Help) dengan halaman yang seharusnya. Hal ini dilakukan pada keempat-toc.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE toc PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp TOC Version 2.0//EN"
"http://java.sun.com/products/javahelp/toc_2_0.dtd">
<toc version="2.0">
  <tocitem text="ModulKeempat">
    <tocitem text="About ModulKeempat" target="org.module.keempat.about"/>
    <tocitem text="Hal. 1" target="org.module.keempat.halSatu"/>
  </tocitem>
</toc>
```

Hasil akhir dari langkah sebelumnya adalah sebagai berikut:



4.3.1.2 Pengamatan

1. Ubahlah judul dokumentasi pada keempat-hs.xml.
2. Tambahkan dokumentasi beberapa halaman berupa sub-node dan sub-sub node.
3. Lengkapi dokumentasi yang ada menggunakan gambar berupa screenshot aplikasi.
4. Lengkapi dokumentasi menggunakan link ke halaman lain.

4.4 RESUME

4.4.1 Jurnal Pengamatan

Selesaikan contoh kasus, kemudian lakukanlah persoalan dari praktik mandiri.

5 BAB V MENGGUNAKAN BEANTREEVIEW

5.1 IDENTITAS

Kajian

Komponen Netbeans Platform

Topik

BeanTreeView Module

Kompetensi Utama

1. Mampu membangun sebuah aplikasi desktop menggunakan BeanTreeView
2. Memahami properties dari komponen BeanTreeView

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 1 x 100 menit

Parameter Penilaian

1. Jurnal Pengamatan

5.2 PERTANYAAN PENDAHULUAN

1. Apakah yang dimaksud dengan BeanTreeView?
2. Apakah kegunaan dari BeanTreeView pada NPA?
3. Apa yang dimaksud dengan Node dan AbstractNode?

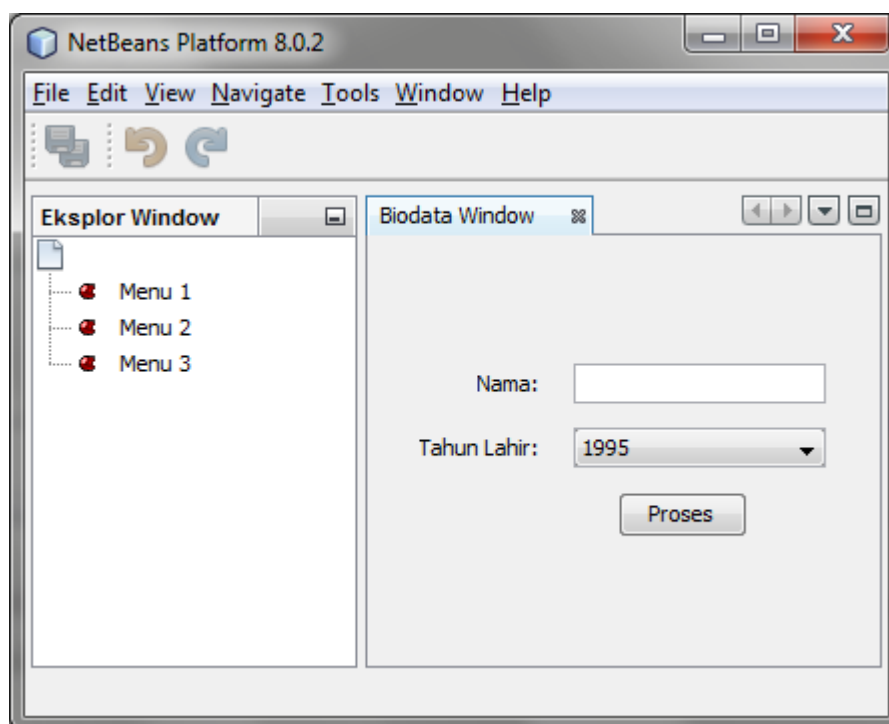
5.3 PRAKTIK

5.3.1 Exercise 1

Pada bagian ini akan dibahas cara mengimplementasikan BeanTreeView dengan menggunakan framework NPA.

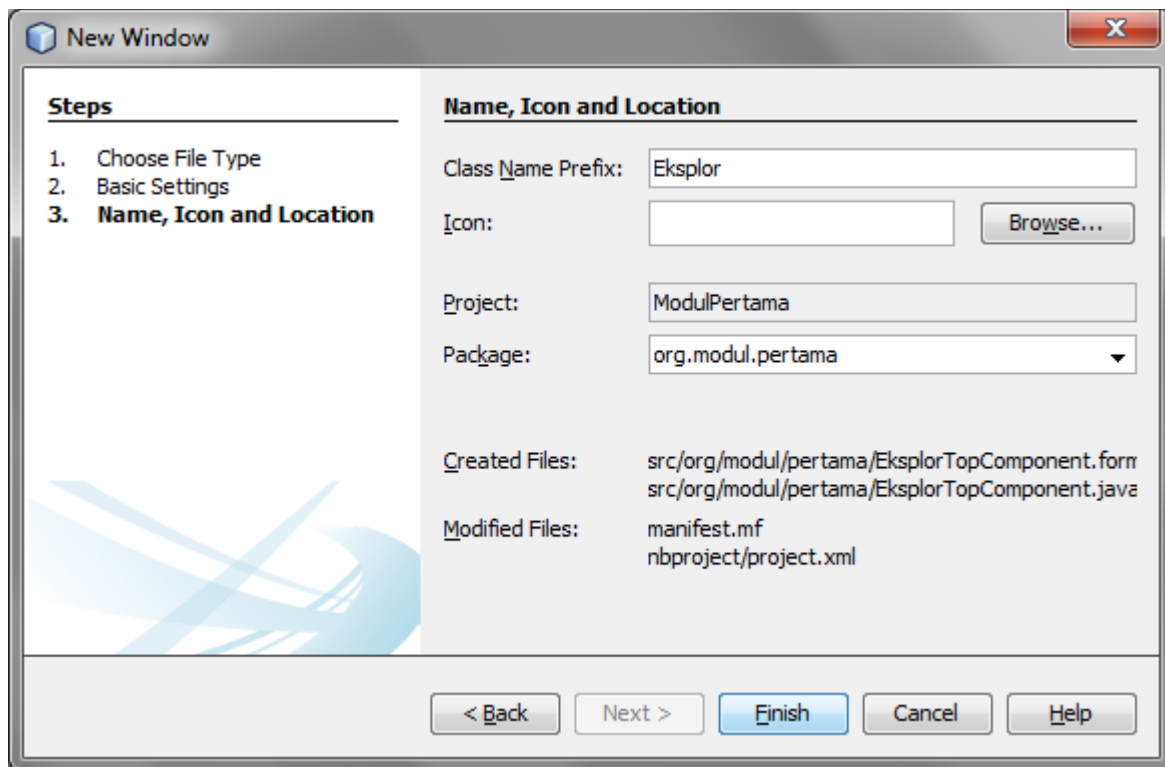
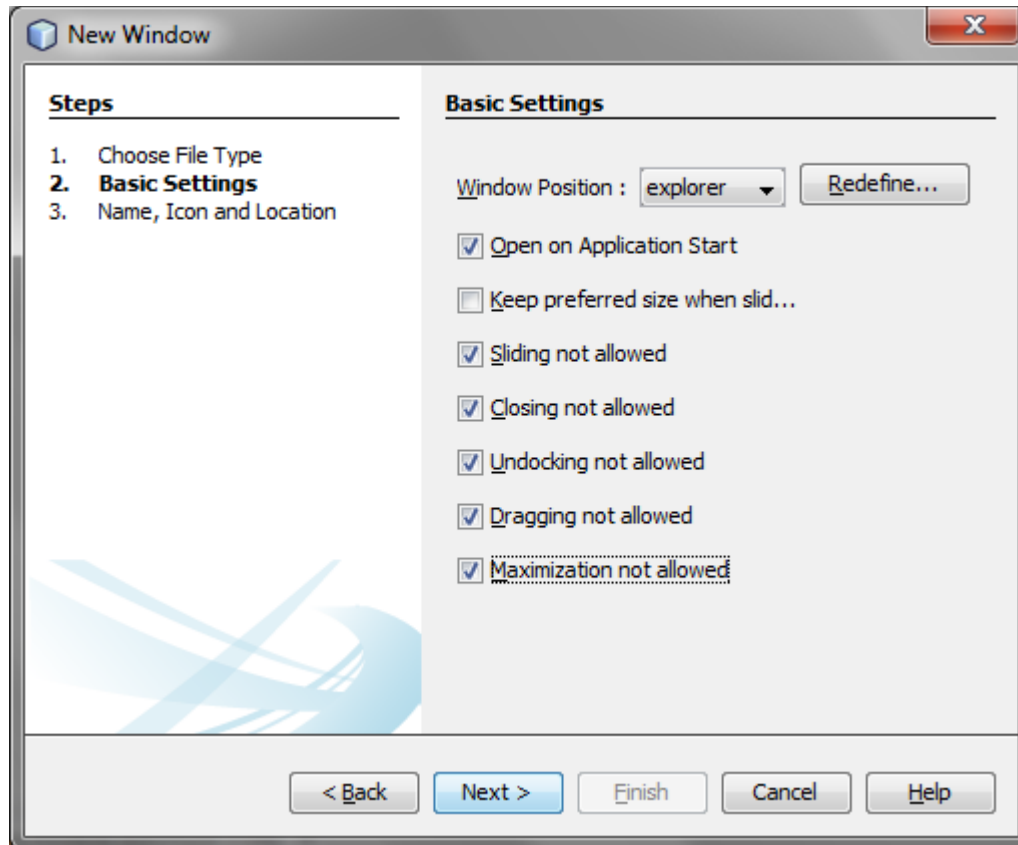
5.3.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap Permasalahan/Soal

Buatlah sebuah Window yang menampilkan Node sebagai berikut:

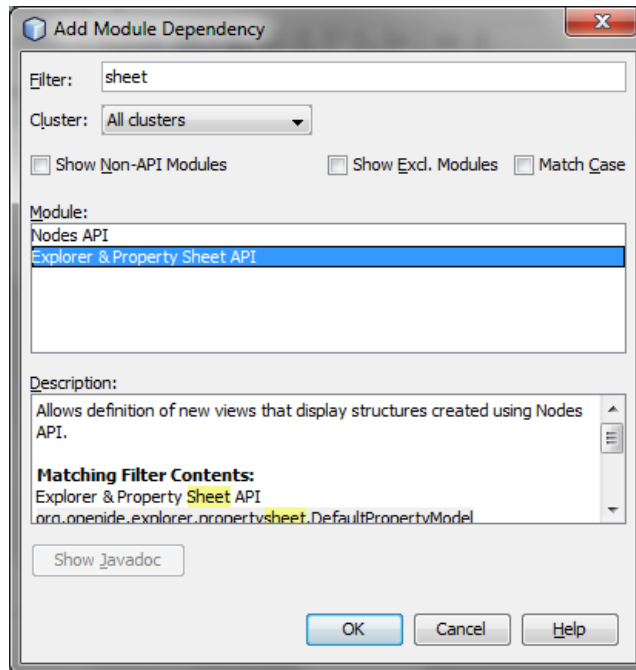


Solusi

Mengacu pada modulPertama, buat sebuah Window dengan konfigurasi sebagai berikut:



Atur di form kemunculan BeanTreeView. Ambil BeanTreeView dari Palette. Jika tidak ada, pergi ke Palette Manager, pilih “Add from JAR”, cari jar “org-openide-explorer.jar” di folder /platform/modules dari direktori home netbeans. Tambahkan pada bagian libraries “Explorer & Property Sheet API” & Nodes API.



Bentuk sebuah class untuk membentuk child atau node2 yang bisa dieksplor nantinya. Bentuk dasar dapat berupa class composite ataupun tipe standar seperti String.

```
import java.beans.IntrospectionException;
import java.util.ArrayList;
import java.util.List;
import org.openide.nodes.BeanNode;
import org.openide.nodes.ChildFactory;
import org.openide.nodes.Node;
import org.openide.util.Exceptions;

public class Simpulanak extends ChildFactory<String> {

    @Override
    protected boolean createKeys(List toPopulate) {
        ArrayList al = new ArrayList();
        al.add("Menu 1");
        al.add("Menu 2");
        al.add("Menu 3");
        toPopulate.addAll(al);
        return true;
    }

    @Override
    protected Node createNodeForKey(String key) {
        try {
            BeanNode bn = new BeanNode(key);

```

```

        bn.setDisplayName(key);
        System.out.println();
        return bn;
    } catch (IntrospectionException ex) {
        Exceptions.printStackTrace(ex);
        return null;
    }
}

```

Pada Window component, tambahkan signature berikut:

```

public final class EksplorTopComponent extends TopComponent implements
ExplorerManager.Provider{

```

Modifikasi override method dari ExplorerManager.Provider sebagai berikut:

```

private final ExplorerManager mgr = new ExplorerManager();
@Override
public ExplorerManager getExplorerManager() {
    return mgr;
}

```

Tambahkan kode berikut pada konstruktor:

```

...
    associateLookup(ExplorerUtils.createLookup(mgr, getActionMap()));
    SimpulAnak ecv = new SimpulAnak();
    AbstractNode an = new AbstractNode(Children.create(ecv, true));
    mgr.setRootContext(an);
...

```

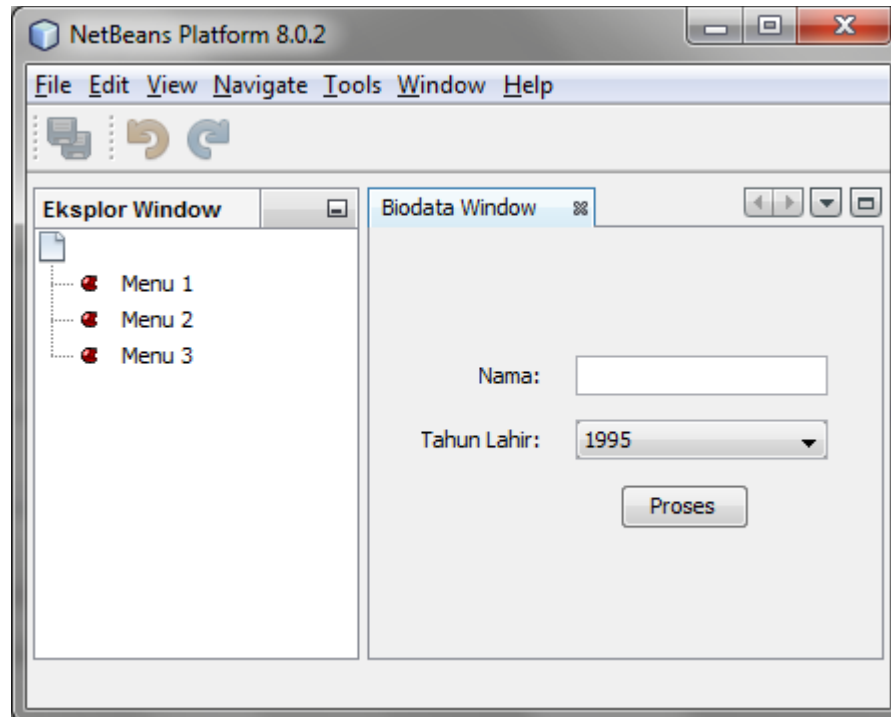
Dan import package sebagai berikut:

```

import org.openide.explorer.ExplorerUtils;
import org.openide.nodes.AbstractNode;
import org.openide.nodes.Children;

```

Hasil dari kode di atas adalah sebagai berikut:



5.3.1.2 *Praktik Mandiri*

1. Tambahkan beberapa node dari node yang sudah terbentuk.
2. Jika ingin membentuk sub-node dari node yang sudah ada, hal apa yang harus dilakukan?
3. Berikan aksi pada setiap node, dan hubungkan sebuah node dengan komponen Window yang tersedia.

5.4 RESUME

5.4.1 **Jurnal Pengamatan**

Selesaikan contoh kasus, kemudian jawablah pertanyaan-pertanyaan yang terdapat pada bagian praktik mandiri.

6 BAB VI KONEKSI JAVA PERSISTENCE API (JPA)

6.1 IDENTITAS

Kajian

Pengaksesan Database

Topik

Menggunakan JPA

Kompetensi Utama

1. Mampu membangun sebuah aplikasi desktop menggunakan komponen dari framework
2. Mampu menghubungkan aplikasi dengan database menggunakan koneksi JPA

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 2 x 100 menit

Parameter Penilaian

1. Jurnal Pengamatan

6.2 PERTANYAAN PENDAHULUAN

1. Apakah yang dimaksud dengan JPA?
2. Apakah fungsi dari JPA tersebut?
3. Bagaimana cara kerja dari JPA?

6.3 PRAKTIK

6.3.1 Exercise 1

Pada bagian ini akan dibahas cara mengimplementasikan JPA.

6.3.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap Permasalahan/Soal

Buatlah sebuah aplikasi java sederhana yang dapat mengakses tabel berikut, dengan menggunakan JPA:

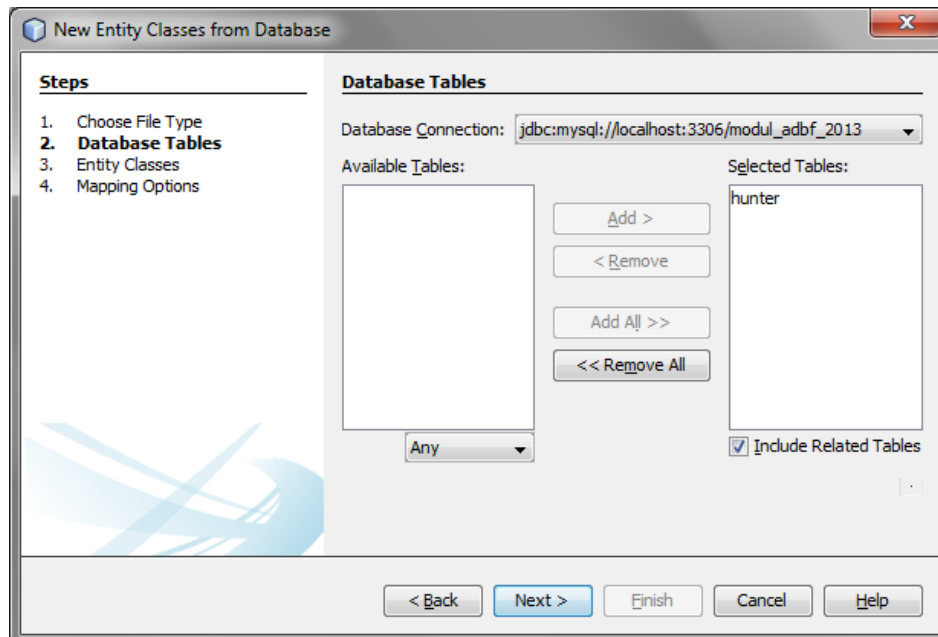
id	nama	thn_lahir	thn_lulus
H-001	Gon Freecs	2000	2010
H-002	Killua Zoldyck	2000	2010

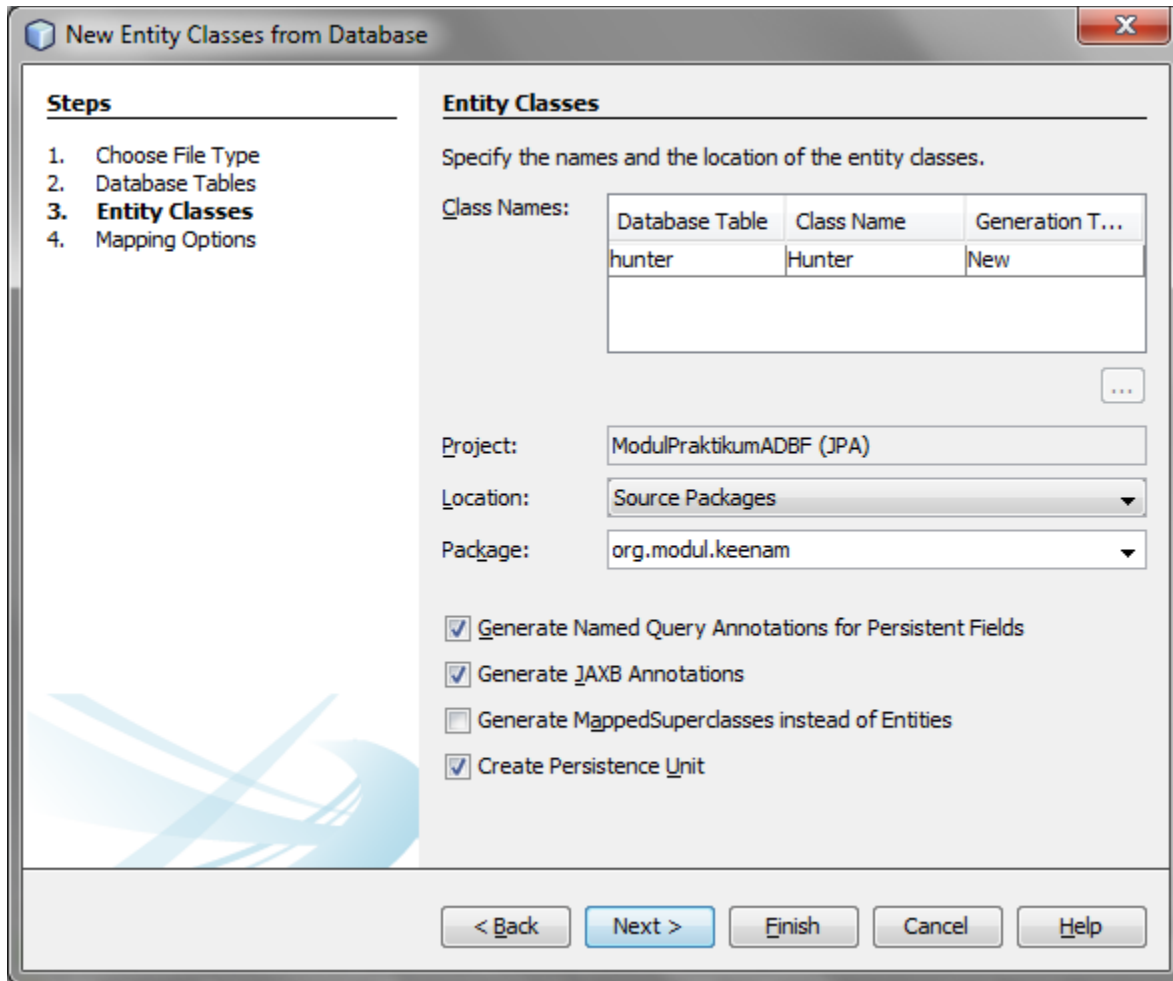
Informasi tambahan dari tabel di atas adalah sebagai berikut:

Database: MySQL
Nama Tabel: Hunter
Nama DB: modul_adbf_2013

Langkah Penyelesaian

Bentuk sebuah Project Netbeans baru, dan bentuk sebuah entitas class (Entity Class From Database) pada project tersebut.





Perhatikan bahwa terbentuk satu xml bernama "persistence.xml". Isinya sebagai berikut:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence          version="2.1"          xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="ModulPraktikumADBF_JPA_PU" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>org.modul.keenam.Hunter</class>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/modul_adbf_2013"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.password" value=""/>
    </properties>
  </persistence-unit>
</persistence>
```

Selain itu terbentuk satu class sebagai berikut:

```
package org.modul.keenam;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.xml.bind.annotation.XmlRootElement;

@Entity
@Table(name = "hunter")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Hunter.findAll", query = "SELECT h FROM Hunter h"),
    @NamedQuery(name = "Hunter.findById", query = "SELECT h FROM Hunter h WHERE h.id = :id"),
    @NamedQuery(name = "Hunter.findByName", query = "SELECT h FROM Hunter h WHERE h.nama = :nama"),
    @NamedQuery(name = "Hunter.findByThnLahir", query = "SELECT h FROM Hunter h WHERE h.thnLahir = :thnLahir"),
    @NamedQuery(name = "Hunter.findByThnLulus", query = "SELECT h FROM Hunter h WHERE h.thnLulus = :thnLulus")})
public class Hunter implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id")
    private String id;
    @Basic(optional = false)
    @Column(name = "nama")
    private String nama;
    @Basic(optional = false)
    @Column(name = "thn_lahir")
    private int thnLahir;
    @Basic(optional = false)
    @Column(name = "thn_lulus")
    private int thnLulus;

    public Hunter() {
    }

    public Hunter(String id) {
        this.id = id;
    }
}
```



```
}

public Hunter(String id, String nama, int thnLahir, int thnLulus) {
    this.id = id;
    this.nama = nama;
    this.thnLahir = thnLahir;
    this.thnLulus = thnLulus;
}

public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getNama() {
    return nama;
}

public void setNama(String nama) {
    this.nama = nama;
}

public int getThnLahir() {
    return thnLahir;
}

public void setThnLahir(int thnLahir) {
    this.thnLahir = thnLahir;
}

public int getThnLulus() {
    return thnLulus;
}

public void setThnLulus(int thnLulus) {
    this.thnLulus = thnLulus;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (id != null ? id.hashCode() : 0);
    return hash;
}
```

```

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Hunter)) {
        return false;
    }
    Hunter other = (Hunter) object;
    if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "org.modul.keenam.Hunter[ id=" + id + " ]";
}
}

```

Untuk menggunakan class (JPA) di atas, bentuk sebuah class sebagai berikut:

```

package org.modul.keenam;

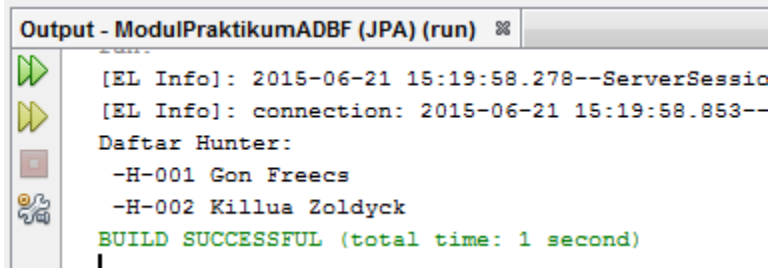
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;
import javax.persistence.Query;

public class AksesHunter {
    public void lihat(){
        EntityManager em = Persistence.
            createEntityManagerFactory("ModulPraktikumADBF__JPA_PU").
            createEntityManager();
        Query kueri = em.createNamedQuery("Hunter.findAll");
        List<Hunter> l = kueri.getResultList();
        System.out.println("Daftar Hunter: ");
        for(Hunter h:l){
            System.out.println("-"+h.getId()+" "+h.getNama());
        }
    }
}

```

```
public static void main(String[] args) {  
    AksesHunter ah = new AksesHunter();  
    ah.lihat();  
}  
}
```

Tambahkan JDBC Driver, dan run program di atas. Hasilnya adalah sebagai berikut:



```
Output - ModulPraktikumADBF (JPA) (run) x  
[EL Info]: 2015-06-21 15:19:58.278--ServerSessio  
[EL Info]: connection: 2015-06-21 15:19:58.853--  
Daftar Hunter:  
-H-001 Gon Freecs  
-H-002 Killua Zoldyck  
BUILD SUCCESSFUL (total time: 1 second)
```

6.3.1.2 *Praktik Mandiri*

1. Perhatikan kode pengaksesan query `Hunter.findAll`, dari mana query tersebut berasal? Gunakan query versi lain yang sesuai.
2. Tambahkan kode untuk melakukan *input-update*, juga *delete* terhadap data yang digunakan.

6.4 RESUME

6.4.1 **Jurnal Pengamatan**

Selesaikan contoh kasus, kemudian jawablah pertanyaan-pertanyaan yang terdapat pada bagian praktik mandiri.

7 BAB VII KOMPONEN PELENGKAP

7.1 IDENTITAS

Kajian

Pengaksesan Database

Topik

Menggunakan Komponen NPA

Kompetensi Utama

1. Mampu membangun sebuah aplikasi desktop menggunakan komponen dari framework
2. Mampu menghubungkan aplikasi dengan database menggunakan koneksi JPA

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 0 x 20 menit
2. Kegiatan Mandiri : 1 x 100 menit

Parameter Penilaian

1. Jurnal Pengamatan

7.2 PRAKTIK

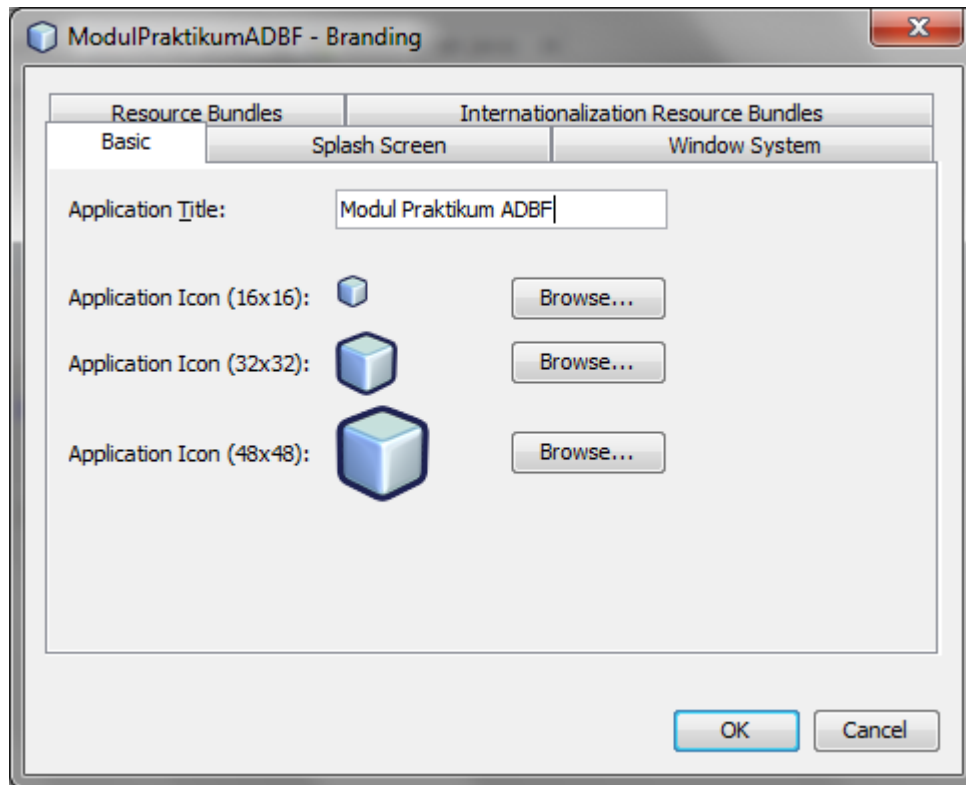
7.2.1 Exercise 1

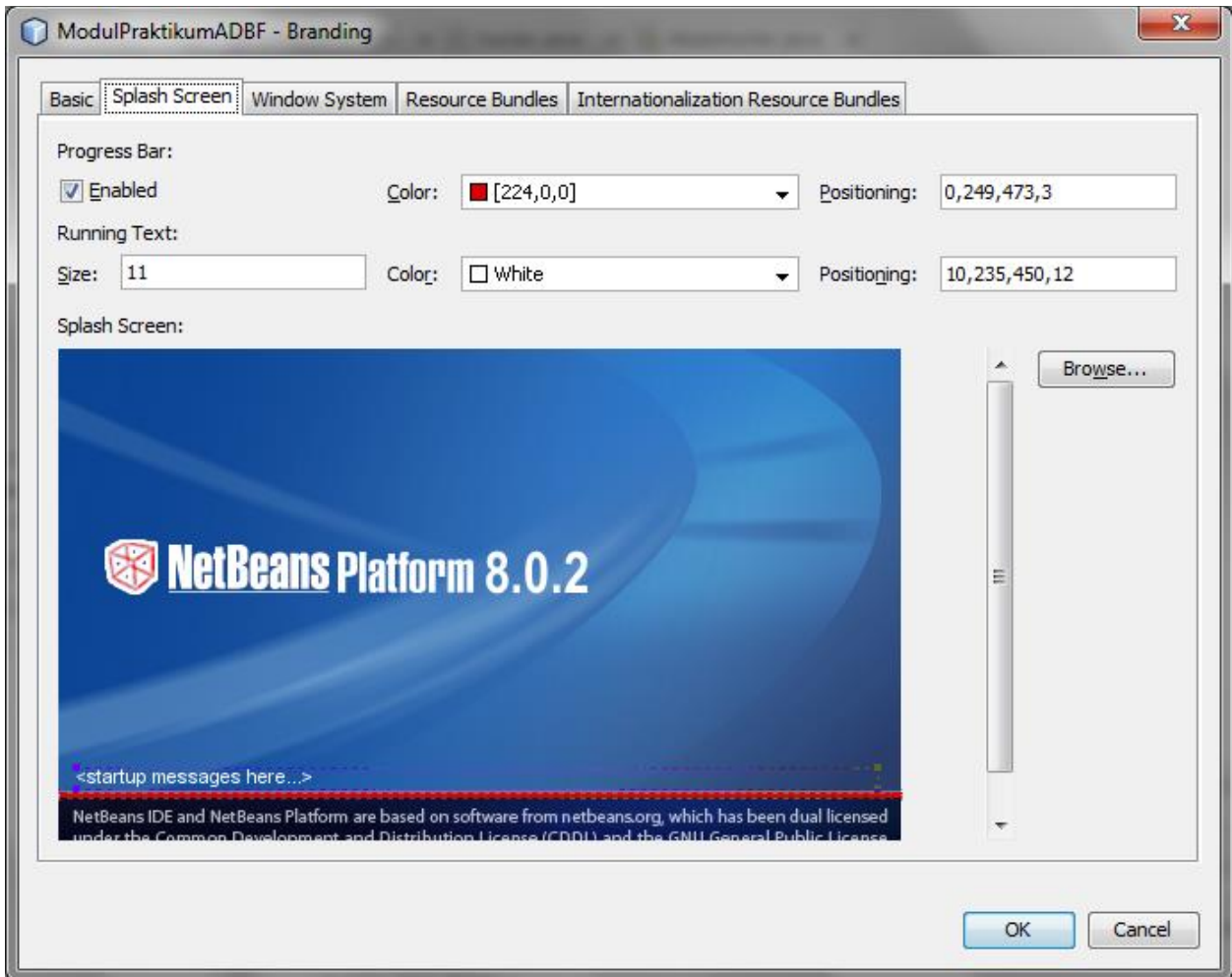
Berdasarkan NPA yang sudah dibentuk sebelumnya, lakukan perubahan terhadap splash screen, dan bentuk installer-nya.

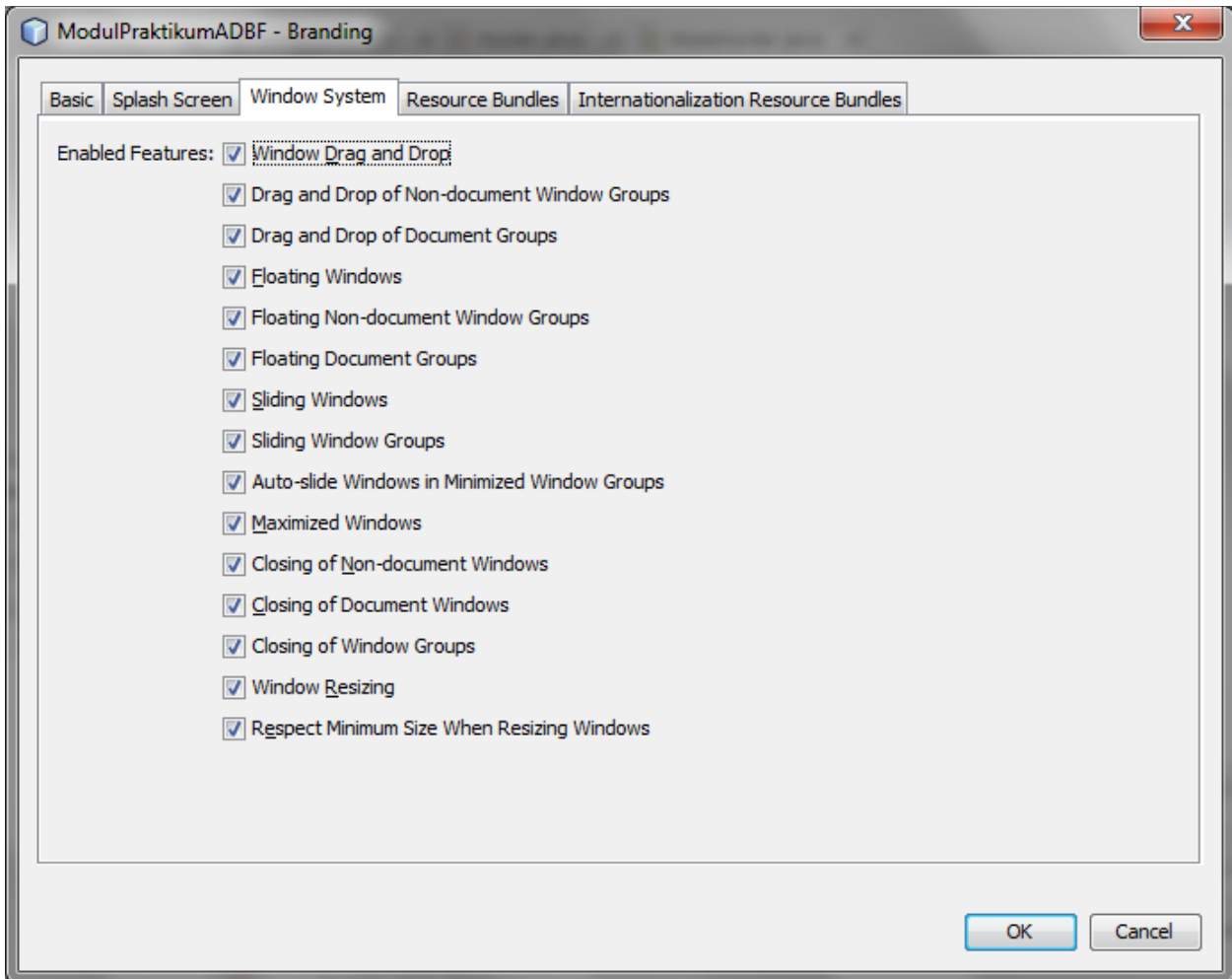
7.2.1.1 Soal, Langkah Penyelesaian dan Solusi Lengkap

Penyelesaian

Pengubahan splash screen dapat dilakukan dengan mengatur branding bagi NPA. Klik kanan NPA, pilih Branding. Terdapat beberapa pilihan yang dapat dilakukan.



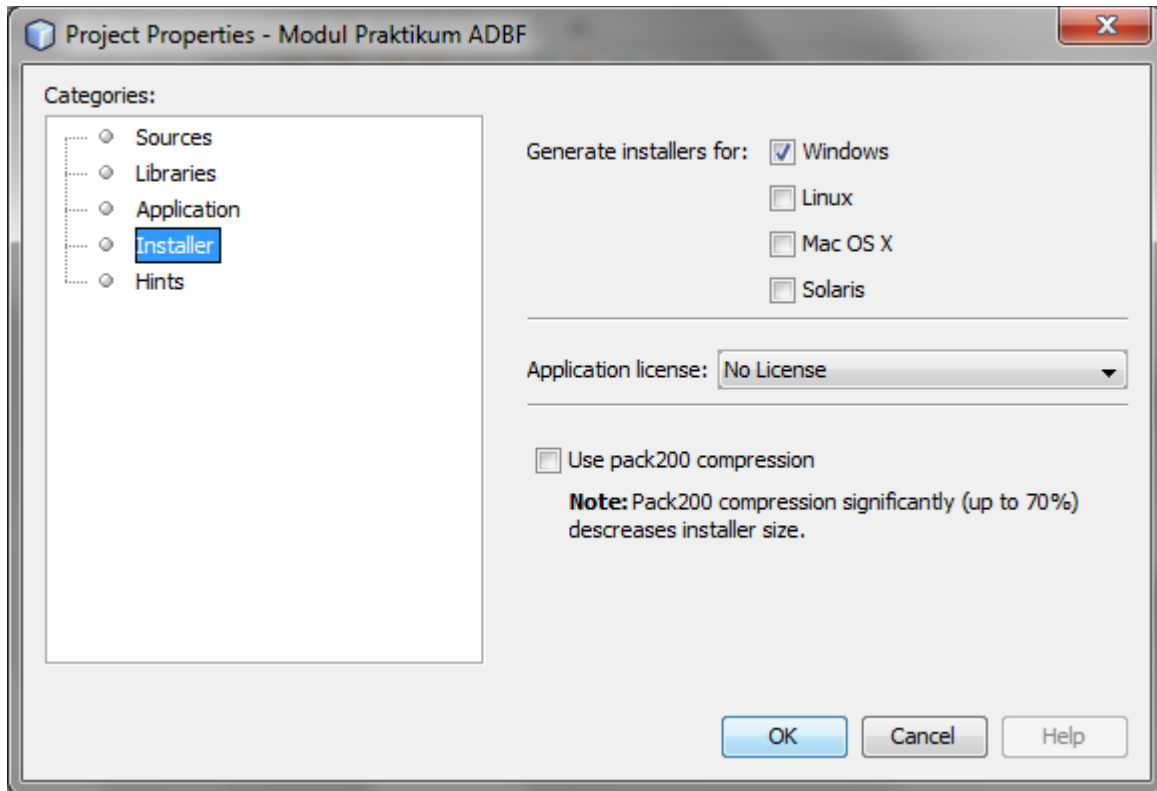




Title bar juga dapat diubah dengan menggunakan kode berikut:

```
Frame f = WindowManager.getDefault().getMainWindow();  
f.setTitle("My title");
```

Setelah melakukan perubahan, lakukan generate installer dengan mengakses properties.



Setelah melakukan pemilihan, klik kanan NPA, pilih “Package as” → “Installers”. Installer yang terbentuk berjalan pada JVM. Pastikan telah meng-install JRE sebelum melakukan instalasi installer NPA. Kesalahan peng-*install*-an dikarenakan belum meng-*install* JRE, dapat ditangani dengan melakukan *bundling* installer dengan JRE.

7.2.2 Jurnal Pengamatan

Selesaikan contoh kasus.

DAFTAR PUSTAKA

[1] Bock, Heiko. 2010. The Definitive Guide to Netbeans Platform 7. Apress.

[2] Netbeans.org